

# TSPL

# 编程手册

Rev.1.0

厦门汉印电子技术有限公司

地址：厦门市湖里区高崎南十二路艾德航空工业园 8 号楼 5 楼

电话：0592-5885991

网址：[www.hpvt.com](http://www.hpvt.com)

## 更改记录

[illegible]

# 目 录

1. 系统设定指令.....	1
SIZE.....	1
GAP.....	2
BLINE.....	3
OFFSET.....	4
SPEED.....	5
DENSITY.....	6
DIRECTION and Mirror Image.....	6
REFERENCE.....	7
SHIFT.....	8
CODEPAGE.....	9
CLS.....	10
FEED.....	10
BACKFEED & BACKUP.....	10
FORMFEED.....	11
HOME.....	12
PRINT.....	12
SOUND.....	13
CUT.....	14
LIMITFEED.....	14
SELFTEST.....	15
2. 标签格式指令.....	16
BAR.....	16
BARCODE.....	17
BITMAP.....	21
BOX.....	23
CIRCLE.....	24
DMATRIX.....	24
ERASE.....	25
MAXICODE.....	26
PDF417.....	27
AZTEC.....	31
PUTBMP.....	33
PUTPCX.....	34
QRCODE.....	35
RSS.....	39
REVERSE.....	40
TEXT.....	41
3. 询问打印机状态指令 (RS-232).....	43

<ESC>!?	43
<ESC>!R	44
~!@	44
~!A	44
~!C	44
~!D	44
~!F	44
~!I	44
~!T	44
4. 文件管理指令	45
DOWNLOAD	45
EOP	47
FILES	47
KILL	48
MOVE	49
RUN	49
5. BASIC 指令及函数	50
ABS()	50
ABC()	50
CHR\$( )	51
END	52
EOF()	53
OPEN	54
WRITE	56
SEEK	58
LOF()	59
FREAD\$( )	60
FOR...NEXT LOOP	61
IF...THEN...ELSE...ENDIF LOOP	62
GOSUB...RETURN	65
GOTO	65
INP\$( )	66
REM	67
OUT	68
GETKEY( )	68
INT( )	69
LEFT\$( )	70
LEN( )	70
MID\$( )	71
RIGHT\$( )	72
STR\$( )	72
STRCOMP ( )	73
INSTR ( )	74

TRIM\$ ( ).....	75
LTRIM\$ ( ).....	75
RTRIM\$ ( ).....	76
VAL ( ).....	77
6. 设备重置指令.....	78
SET COUNTER.....	78
SET PARTIAL_CUTTER.....	79
SET BACK.....	80
SET KEY1, SET KEY2, SET KEY3.....	81
SET LED1, SET LED2, SET LED3.....	82
SET PEEL.....	83
SET TEAR & SET STRIPER.....	84
SET GAP.....	84
SET RIBBON.....	85
SET COM1.....	86
SET PRINTKEY.....	87
SET REPRINT.....	87
PEEL.....	88
LED1, LED2, LED3.....	89
7. 打印机全域变数.....	91
@LABEL.....	91
YEAR.....	92
MONTH.....	93
DATE.....	94
WEEK.....	95
HOUR.....	96
MINUTE.....	97
SECOND.....	98
@YEAR.....	99
@MONTH.....	99
@DATE.....	100
@DAY.....	101
@HOUR.....	101
@MINUTE.....	102
@SECOND.....	103

# 1. 系统设定指令

## SIZE

该指令用于设定标签的宽度和长度。

### 指令语法

1) 英制系统（英寸）

SIZE m, n

2) 公制系统（毫米）

SIZE m mm, n mm

3) 以点 dots 表示

SIZE m dot, n dot

### 参数

### 说明

m

标签宽度（英寸或毫米）

n

标签长度（英寸或毫米）

### 备注：

200 DPI : 1 mm = 8 dots

300 DPI : 1mm = 12 dots

对于公制系统和以点 dots 表示的系统，参数与单位（“毫米”或“点”）之间必须有空格。

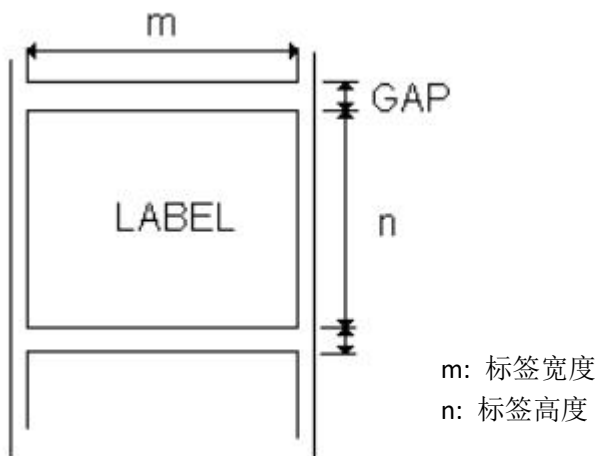
### 示例

(1) 英制系统（英寸）

SIZE 3.5, 3.00

(2) 公制系统（毫米）

SIZE 100 mm, 100 mm



## GAP

该指令用于设定两个标签之间的垂直距离。

### 指令语法

1) 英制系统（英寸）

GAP m, n

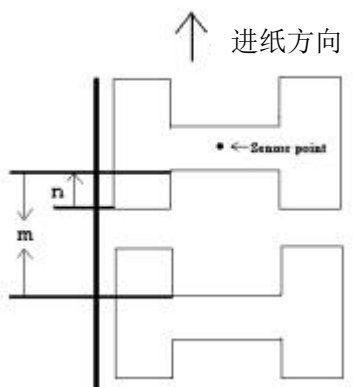
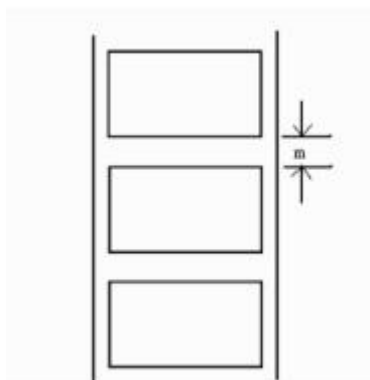
2) 公制系统（毫米）

GAP m mm, n mm

### 参数

### 说明

m	两个标签之间的垂直距离 $0 \leq m \leq 1 (\text{inch}), 0 \leq m \leq 25.4 (\text{mm})$
n	垂直间距的偏移 $n \leq \text{标签长度 (英寸或毫米)}$
0,0	连续标签



### 备注:

对于公制系统，参数与单位“毫米”之间必须有空格。

当传感器类型由“黑标”变为“缝标”，请先把“GAP”指令发送给打印机。

其他：在 DOS 模式下，**DOS模式暂不支持，可删去**

C:\>copy con lpt1 <Enter>

GAP 2 mm,0 <Enter>

<Ctrl>+<Z> <Enter>

## 示例

### 一般垂直间距设定

#### 1) 英制系统（英寸）

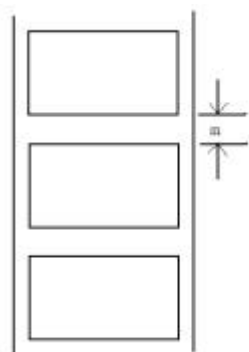
GAP 0.12, 0

#### 2) 公制系统（毫米）

GAP 3 mm, 0

#### 3) 连续标签

GAP 0, 0



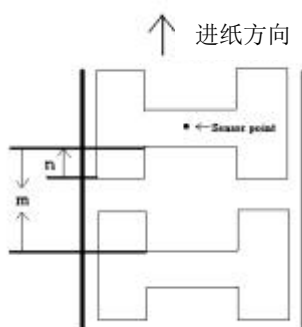
### 特殊垂直间距设定

#### 1) 英制系统（英寸）

GAP 0.30, 0.10

#### 2) 公制系统（毫米）

GAP 7.62 mm, 2.54 mm



## BLINE

该指令用于设置黑线的高度和用户自定义的额外的标签进纸长度。

### 指令语法

#### 1) 英制系统（英寸）

BLINE m, n

#### 2) 公制系统（毫米）

BLINE m mm, n mm

### 参数

### 说明

m

两个黑标之间的垂直距离（英寸或毫米）

$0 \leq m \leq 1 \text{ (inch)}, 0 \leq m \leq 25.4 \text{ (mm)}$

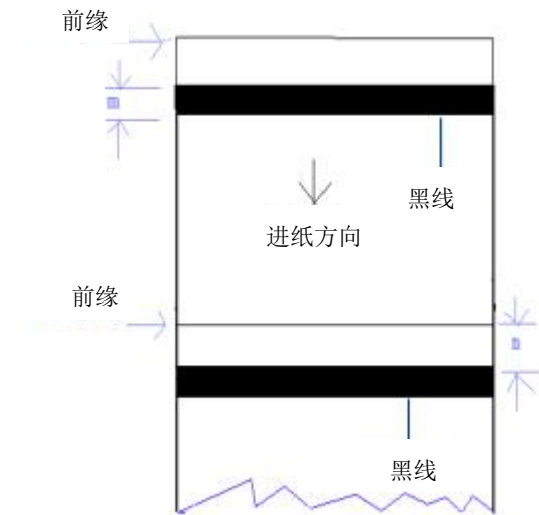
n

额外标签的进纸长度  $0 \leq n \leq \text{标签长度}$

0,0

连续标签





### 备注:

对于公制系统，参数与单位“毫米”之间必须有空格。

当传感器类型由“缝标”变为“黑标”，请先把“BLINE”指令发送给打印机。

其他：在 DOS 模式下，

```
C:\>copy con lpt1 <Enter>
```

```
    GAP 2 mm,0 <Enter>
```

```
    <Ctrl>+<Z> <Enter>
```

### 示例

1) 英制系统（英寸）

```
BLINE 0.20, 0.50
```

2) 公制系统（毫米）

```
BLINE 5.08 mm, 12.7 mm
```

## OFFSET

该指令用于控制剥离模式和切刀模式下打印机定位完后再走纸的距离。

### 指令语法

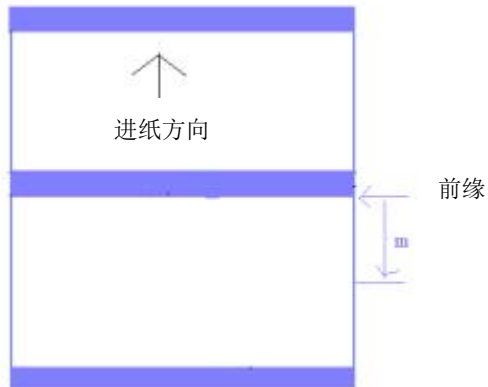
1) 英制系统（英寸）

```
OFFSET m
```

2) 公制系统（毫米）

```
OFFSET m mm
```

参数	说明
m	偏移的距离（英寸或毫米） $-1 \leq m \leq 1(\text{inch})$



**注意：**不适当的偏移值可能引起卡纸。

#### 示例

1) 英制系统（英寸）

OFFSET 0.5

2) 公制系统（毫米）

OFFSET 12.7 mm

## SPEED

该指令用于设置打印速度。

#### 指令语法

SPEED n

参数	说明
n	打印速度 英寸/秒

#### 示例

SPEED 10

## DENSITY

该指令用于设置打印浓度。

### 指令语法

DENSITY n

#### 参数

n

#### 说明

0~15

0: 最淡的打印浓度

15: 最深的打印浓度

备注: 默认浓度设置为8。

### 示例

DENSITY 7

## DIRECTION and Mirror Image

该指令用于定义打印机的打印方向和镜像，且此设置将会被存储在打印机内存里。

### 指令语法

DIRECTION n[,m]

#### 参数

n

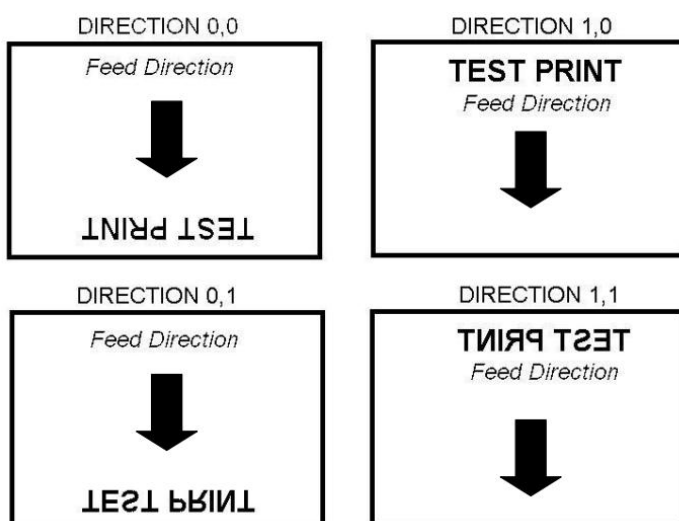
#### 说明

0 或 1.请参考下图

m

0: 打印常规图片

1: 打印镜像图片



## 示例

DIRECTION 0

DIRECTION 0,1

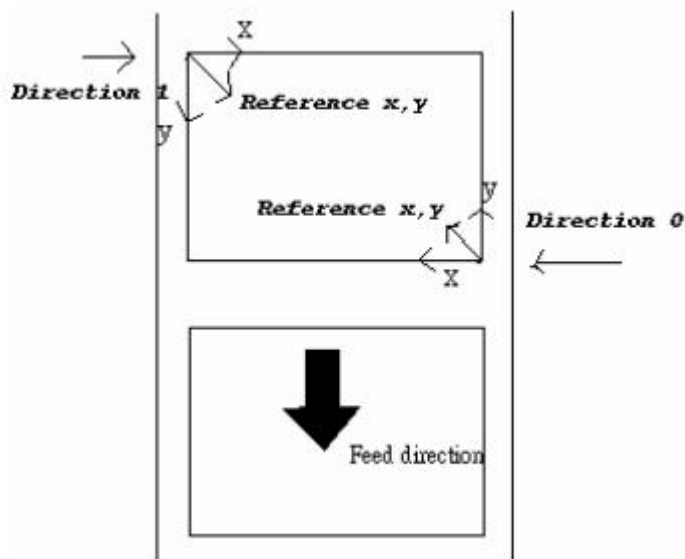
## REFERENCE

该指令用于定义标签的参考起点。参考点（原点）因打印方向而异，如下所示：

### 指令语法

REFERENCE x, y

参数	说明
x	横坐标（以点 dots 表示）
y	纵坐标（以点 dots 表示）



备注：200 DPI: 1mm = 8 dots

300 DPI: 1mm = 12 dots

## 示例

REFERENCE 10,10

## SHIFT

该指令用于移动标签的垂直位置。正值是让标签进一步远离打印方向；负值是将标签移至打印方向。

### 指令语法

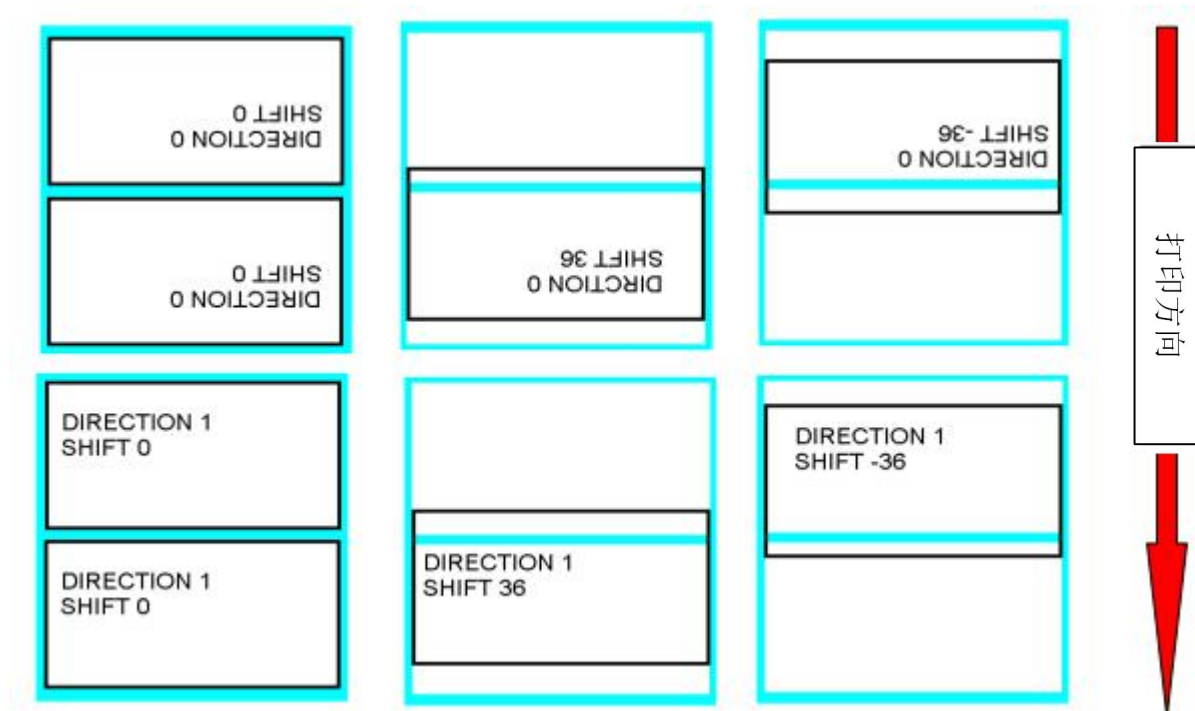
SHIFT n

### 参数

### 说明

n 最大值为 1 英寸。对于 200 dpi 的打印机，范围为-203 到 203；对于 300 dpi 的打印机，范围为-300 到 300。单位为点(dots)。

### 示例



SIZE 4,2.5

GAP 2 mm,0

DIRECTION 0

SHIFT 30

OFFSET 0

CLS

TEXT 400,200,"3",0,1,1,"DIRECTION 0"

TEXT 400,250,"3",0,1,1,"SHIFT 30"

BOX 10,0,780,490,8

PRINT 3,1

## CODEPAGE

该指令用于定义国际字符集的代码。

### 指令语法

CODEPAGE n

#### 参数

#### 说明

n 代码页的名称或数字，可以分成 7 字节代码页和 8 字节代码页  
7 字节代码页名称

USA: USA

BRI: British

GER: German

FRE: French

DAN: Danish

ITA: Italian

SPA: Spanish

SWE: Swedish

SWI: Swiss

8 字节代码页数字

437: United States

850: Multilingual

852: Slavic

860: Portuguese

863: Canadian/French

865: Nordic

857: Turkish

Windows 代码页

1250: Central Europe

1252: Latin I

1253: Greek

1254: Turkish

备注：数据长度决定了通讯的参数是 7 字节还是 8 字节。

### 示例

CODEPAGE 437

## CLS

该指令用于清除图像缓冲区。

### 指令语法

CLS

备注：该指令必须放于 SIZE 指令之后。

### 示例

CLS

## FEED

该指令指定标签进纸的长度。长度以点 dots 表示。

### 指令语法

FEED n

### 参数

### 说明

n

unit: dot

$1 \leq n \leq 9999$

备注： **200 DPI: 1 mm = 8 dots**

**300 DPI: 1 mm = 12 dots**

### 示例

FEED 40

## BACKFEED & BACKUP

该指令用于设置反向进纸。进纸长度以点 dots 表示。

### 指令语法

BACKUP n

BACKFEED n

### 参数

### 说明

n

unit: dot

$1 \leq n \leq 9999$

备注: **200 DPI: 1 mm = 8 dots**  
**300 DPI: 1 mm = 12 dots**

注意: 不适当的后进纸值可能会造成卡纸或褶皱。

示例

BACKUP 40

BACKFEED 40

## FORMFEED

该指令用于将标签定位到下一张标签的起始处。

指令语法

FORMFEED

示例

SIZE 4, 2.5

GAP 0 mm,0

SPEED 4

DENSITY 7

DIRECTION 0

OFFSET 0.00

REFERENCE 0,0

SET PEEL OFF

SET CUTTER OFF

SET COUNTER @0 +1

@0="000001"

FORMFEED

CLS

BOX 1,1,360,65,12

TEXT 25,25, "3",0,1,1,"FORMFEED COMMAND TEST"

TEXT 25,80,"3",0,1,1,@0

PRINT 3,1

gap 0mm会设置纸张类型为连续纸，  
标签纸 例：  
gap 2 mm , 0 ( 缝隙为2mm )



## HOME

该指令用于将标签定位到内部传感器确定的起点。使用此指令之前必须指定标签的尺寸和缝隙。

### 指令语法

HOME

### 示例

```
SIZE 4,2.5
GAP 2 mm,0
SPEED 4
DENSITY 7
DIRECTION 0
OFFSET 0.00
REFERENCE 0,0
SET PEEL OFF
SET CUTTER OFF
SET COUNTER @0 +1
@0="000001"
HOME
CLS
BOX 1,1,360,65,12
TEXT 25,25,"3",0,1,1,"HOME COMMAND TEST"
TEXT 25,80,"3",0,1,1,@0
PRINT 3,1
```

## PRINT

该指令用于打印当前存储在图像缓冲区的标签。

### 指令语法

PRINT m [,n]

### 参数

m

### 说明

指定打印多少个标签。

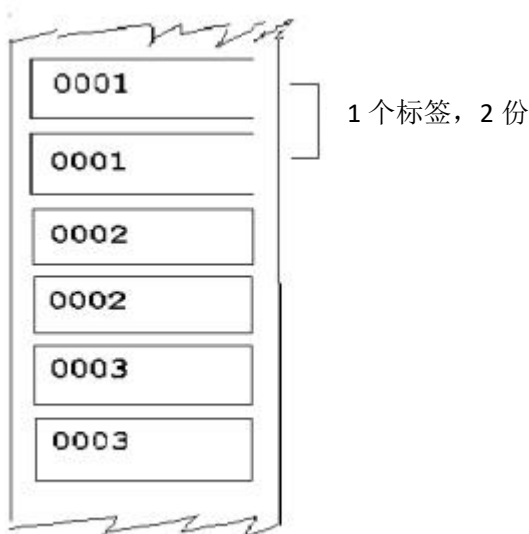
$1 \leq m \leq 999999999$

如果  $m=-1$ ，打印机将打印多份最后一张标签的内容。 $m<1$ ，不合法，打印机不作处理

n 指定每张特定的标签要打印多少份。  $1 \leq n \leq 999999999$

### 示例

```
SIZE 60 mm, 20 mm
SET COUNTER @1 1
@1="0001"
CLS
TEXT 10,10,"3",0,1,1,@1
PRINT 3,2
PRINT - 1,2
```



## SOUND

该指令用于控制蜂鸣器的声音频率，可设定 10 阶的声音。每个声音的长短由“interval（时间长度）”参数控制。

### 指令语法

SOUND level, interval

#### 参数

音阶 level

时间长度 interval

#### 说明

声音等级：0~9

声间持续时间：1~4095

### 示例

```
SOUND 5,200
SOUND 3,200
SOUND 3,200
SOUND 4,200
SOUND 2,200
SOUND 2,200
SOUND 1,200
SOUND 2,200
SOUND 3,200
SOUND 4,200
SOUND 5,200
```

## CUT

该指令用于激活切刀在不退纸的情况下立即切纸。

### 指令语法

CUT

### 示例

SIZE 3,3

GAP 0 mm,0

CLS

DIRECTION 0

REFERENCE 0,0

OFFSET 0.00 mm

SET CUTTER OFF

SET TEAR OFF

BOX 0,0,866,866,5

TEXT 100,100,"5",0,1,1,"FEED & CUT"

TEXT 100,200,"5",0,1,1,"300 DPI"

PRINT 1,1

FEED 260

CUT

## LIMITFEED

若进纸时缝标传感器无法侦测到纸张，打印机就无法定位缝标的位置。该指令用于设定打印机进纸时，若经过所设定的长度仍无法侦测到垂直间距，则打印机发生错误，停止进纸。

### 指令语法

LIMITFEED n (英寸，英制系统)

LIMITFEED n mm (毫米，公制系统)

LIMITFEED n dot

### 参数

### 说明

n 传感器侦测到的最大间距

### 备注：

该设置会保留在存储器里。

打印机初始化时默认值为10英寸。

对于英制系统，参数m和n之间必须有空格。

## SELFTEST

该指令用于打印打印机信息。

### 指令语法

SELFTEST

### 示例

SELFTEST

## 2. 标签格式指令

### BAR

该指令用于在标签上画一条线。

#### 指令语法

BAR x, y, width, height

参数	说明
x	左上角 x 坐标（以点 dots 表示）
y	左上角 y 坐标（以点 dots 表示）
width	线宽（以点 dots 表示）
height	线高（以点 dots 表示）

**备注：** 200 DPI: 1 mm = 8 dots  
300 DPI: 1 mm = 12 dots

建议最大的线高为 12 点，宽度为 4 英寸。线高超过 12 点可能会损坏电源影响打印质量。  
最大的印字率因机型而异。桌面打印机和工业打印机的印字率上限分别为 20% 和 30%。

#### 示例

```
SIZE 4,2.5
GAP 0,0
SPEED 6
DENSITY 8
DIRECTION 0
CLS
BAR 100, 100, 300, 200
PRINT 1,1
```



## BARCODE

该指令用于打印一维条码。可用条码如下所示：

Code 128 (自动切换代码子集)  
 Code 128M (手动切换代码子集)  
 EAN 128 (自动切换代码子集)  
 Interleaved 2 of 5  
 Interleaved 2 of 5 with check digit  
 Code 39 standard  
 Code 39 full ASCII  
 Code 39 full ASCII with check digit  
 Code 93  
 EAN 13  
 EAN 13 with 2 digits add-on  
 EAN 13 with 5 digits add-on  
 EAN 8  
 EAN 8 with 2 digits add-on  
 EAN 8 with 5 digits add-on  
 Codabar  
 Postnet  
 UPC-A  
 UPC-A with 2 digits add-on  
 UPC-A with 5 digits add-on  
 UPC-E  
 UPC-E with 2 digits add-on  
 UPC-E with 5 digits add-on  
 MSI  
 PLESSEY  
 China POST  
 ITF14  
 EAN14  
 CODE 11

### 指令语法

BARCODE X, Y, “code type”, height, human readable, rotation, narrow, wide, “code”

参数	说明	[alignment, ]"content"
X	指定标签上的条形码的 X 坐标	
Y	指定标签上的条形码的 Y 坐标	
条码类型		
128	Code 128, 自动转换代码子集 A,B,C	
128M	Code 128, 手动转换代码子集 A,B,C	

Control code	A	B	C
096	FNC3	FNC3	NONE
097	FNC2	FNC2	NONE
098	SHIFT	SHIFT	NONE
099	CODE C	CODE C	NONE
100	CODE B	FNC4	CODE B
101	FNC4	CODE A	CODE A
102	FNC1	FNC1	FNC1
103	Start (CODE A)		
104	Start (CODE B)		
105	Start (CODE C)		

使用“!”作为控制码三位数之后的一个初始字符。如果没有设置起始子集，默认为 B。

height	条码高度（以点 dots 表示）
human readable	0: 不可读 1: 可读
rotation	0: 不旋转 90: 顺时针旋转 90 度 180: 顺时针旋转 180 度 270: 顺时针旋转 270 度
narrow	窄条的宽度（以点 dots 表示）
wide	宽条的宽度（以点 dots 表示）

	narrow : wide 1:1	narrow : wide 1:2	narrow : wide 1:3	narrow : wide 2:5	narrow : wide 3:7
128	10x	-	-	-	-
EAN128	10x	-	-	-	-
25	-	10x	10x	5x	-
25C	-	10x	10x	5x	-
39	-	10x	10x	5x	-
39C	-	10x	10x	5x	-
93	-	-	10x	-	-
EAN13	8x	-	-	-	-
EAN13+2	8x	-	-	-	-
EAN13+5	8x	-	-	-	-
EAN 8	8x	-	-	-	-
EAN 8+2	8x	-	-	-	-
EAN 8+5	8x	-	-	-	-
CODA	-	10x	10x	5x	-
POST	1x	-	-	-	-
UPCA	8x	-	-	-	-
UPCA+2	8x	-	-	-	-
UPCA+5	8x	-	-	-	-
UPCE	8x	-	-	-	-
UPCE+2	8x	-	-	-	-
UPCE+5	8x	-	-	-	-
CPOST	-	-	-	-	1x
MSI	-	-	10x	-	-
MSIC			10x		-
PLESSY	-	-	10x	-	-
ITF14	-	10x	10x	5x	-

EAN14	8x	-	-	-	-
11	-	10x	10x	5x	-



代码序号    条码最大位数

Barcode type	Maximum code length	bar
128	-	
EAN128	-	
25	-	
25C	-	
39	-	
39C	-	
93	-	
EAN13	12	
EAN13+2	14	
EAN13+5	17	
EAN 8	7	
EAN 8+2	9	
EAN 8+5	12	
CODA	-	
POST	5,9,11	
UPCA	11	
UPCA+2	13	
UPCA+5	16	
UPCE	6	
UPCE+2	8	
UPCE+5	11	
CPOST	-	
MSI	-	
MSIC		
PLESSY	-	
ITF14	13	
EAN14	13	
CODE 11	-	

## 示例

BARCODE 100,100," 39" ,96,1,0,2,4," 1000"

BARCODE 10,10," 128M" ,48,1,0,2,2," !104!096ABCD!101EFGH"

(The above example of code 128M encoded with CODE B start character. The next character will be the code 128 function character FNC3 which is then followed by the ABCD characters and EFGH characters encoded as CODE A subset.)

## BITMAP

本指令用于绘制位图图像和 BMP 图像文件。

### 指令语法

BITMAP X, Y, width, height, mode, bitmap data...

#### 参数

#### 描述

X 指定 X 坐标

Y 指定 Y 坐标

width 图像宽度（以比特 bytes 为单位）

height 图像高度（以点 dots 表示）

mode 位图模式如下：

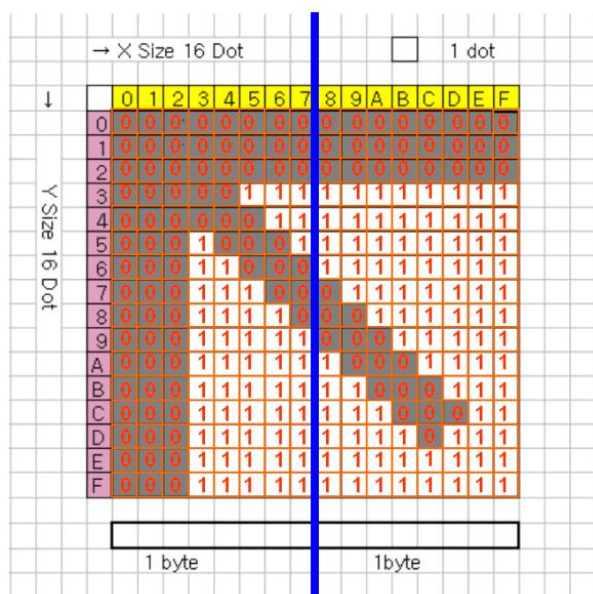
0: OVERWRITE

1: OR

2: XOR

Bitmap data 位图数据

## 示例



ROW (Y- axis)	L-Byte		R-Byte	
	Binary	Hexadecimal	Binary	Hexadecimal
0	00000000	00	00000000	00
1	00000000	00	00000000	00
2	00000000	00	00000000	00
3	00000111	07	11111111	FF
4	00000011	03	11111111	FF
5	00010001	11	11111111	FF
6	00011000	18	11111111	FF
7	00011100	1C	01111111	7F
8	00011110	1E	00111111	3F
9	00011111	1F	00011111	1F
A	00011111	1F	10001111	8F
B	00011111	1F	11000111	C7
C	00011111	1F	11100011	E3
D	00011111	1F	11110111	F7
E	00011111	1F	11111111	FF
F	00011111	1F	11111111	FF

SIZE 4,2

GAP 0,0

CLS

BITMAP 200,200,2,16,0,

PRINT 1,1

L ◀ ↑ -?????

Hexadecimal	ASCII
53 49 5A 45 20 34 2C 32 0D 0A 47 41 50	SIZE 4,2
20 30 2C 30 0D 0A 43 4C 53 0D 0A 42	GAP 0,0
49 54 4D 41 50 20 32 30 30 2C 32 30 30	CLS
2C 32 2C 31 36 2C 30 2C 00 00 00 00 00	BITMAP 200,200,2,16,0,
00 07 FF 03 FF 11 FF 18 FF 1C 7F 1E	L ◀ ↑ -?????
3F 1F 1F 1F 8F 1F C7 1F E3 1F E7 1F	PRINT 1,1
FF 1F FF 0D 0A 50 52 49 4E 54 20 31	
2C 31 0D 0A	

## BOX

该指令用于在标签上绘制矩形。

### 指令语法

BOX X\_start, Y\_start, X\_end, Y\_end, line thickness

参数	描述
X_start	指定左上角的 x 坐标（以点 dots 表示）
Y_start	指定左上角的 y 坐标（以点 dots 表示）
X_end	指定右下角的 x 坐标（以点 dots 表示）
Y_end	指定右下角的 y 坐标（以点 dots 表示）
line thickness	线厚度（以点 dots 表示）

备注: **200 DPI: 1 mm = 8 dots**  
**300 DPI: 1 mm = 12 dots**

矩形的最大厚度建议为 12mm, 4 英寸宽。大于 12mm 的矩形厚度可能会损坏电源影响打印质量。

最大印字率因机型而异。桌面打印机和工业打印机的印字率上限分别为 20% 和 30%。

### 示例

SIZE 4,2.5

GAP 0,0

SPEED 6

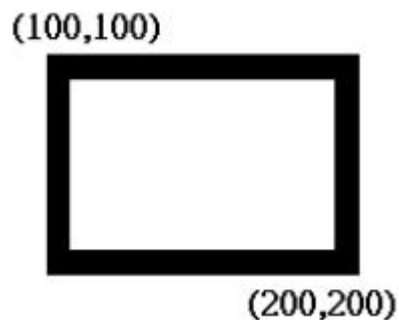
DENSITY 8

DIRECTION 0

CLS

BOX 100,100,200,200,5

PRINT 1,1



## CIRCLE

该指令用于在标签上画圆。

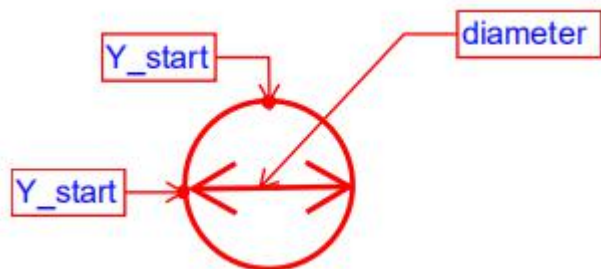
### 指令语法

CIRCLE X\_start, Y\_start, diameter, circle thickness

参数	描述
X_start	指定左上角的 x 坐标（以点 dots 表示）
Y_start	指定左上角的 y 坐标（以点 dots 表示）
diameter	指定圆圈的直径（以点 dots 表示）
thickness	圆圈的厚度（以点 dots 表示）

### 示例

```
SIZE 80 mm, 30 mm
GAP 0,0
DIRECTION 1
CLS
BAR 250,20,100,1
BAR 250,20,1,100
CIRCLE 250,20,100,1
PRINT 1
```



## DMATRIX

该指令用于定义数据点阵二维条码。

### 指令语法

DMATRIX x, y, width, height, [xm,row,col], expression

参数	描述
x	横向起始位置（以点 dots 表示）
y	纵向起始位置（以点 dots 表示）
width	条码区的预期宽度（以点 dots 表示）
height	条码区的预期高度（以点 dots 表示）
xm	模块大小（以点 dots 表示）
row	行高：10 到 144
col	列宽：10 到 144

## 示例

```
SIZE 3,3
GAP 0,0
SPEED 4
DENSITY 8
DIRECTION 0
REFERENCE 0,0
OFFSET 0.00
SET CUTTER OFF
SET TEAR ON
CLS
DMATRIX 10,110,400,400,"DMATRIX EXAMPLE 1"
DMATRIX 310,110,400,400,x6,"DMATRIX EXAMPLE 2"
DMATRIX 10,310,400,400,x8,18,18,"DMATRIX EXAMPLE 3"
PRINT 1,1
```

## ERASE

该指令用于清除图像缓冲区域里的特定区域。

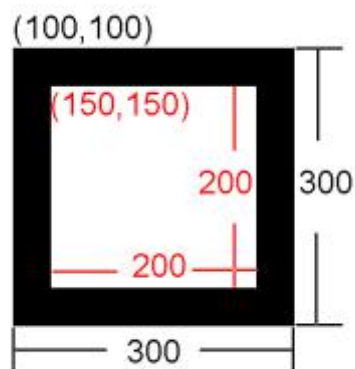
### 指令语法

ERASE X\_start, Y\_start, X\_width, Y\_height

参数	描述
X_start	起点的 x 坐标（以点 dots 表示）
Y_start	起点的 y 坐标（以点 dots 表示）
X_width	x 坐标方向的区域宽度（以点 dots 表示）
Y_height	y 坐标方向的区域高度（以点 dots 表示）

## 示例

```
SIZE 4,2.5
GAP 0,0
SPEED 6
DENSITY 8
DIRECTION 0
CLS BAR 100, 100, 300, 300
ERASE 150,150,200,200
PRINT 1,1
```



## MAXICODE

该指令用于定义 2D Maxicode。

### 指令语法

MAXICODE x, y, mode, [class, country, post, Lm,] "message"

对于模式 2 或 3:

MAXICODE x, y, 模式, 级别, 国家, 邮编, “低优先级信息”

如果国家代码为 840, 邮编的格式为 99999,9999

For other countries, the code is up to 6 alphanumeric characters.

对于其他国家, 代码为最多 6 个英文字母。

对于模式 4,5,6

MAXICODE x, y, mode, [Lm], "message"

参数	描述
x	起点的 X 坐标 (以点 dots 表示)
y	起点的 Y 坐标 (以点 dots 表示)
mode	2, 3, 4, 5
class	服务等级, 3 位数 (针对模式 2, 3)
country	国家代码, 3 位数 (针对模式 2, 3)
post	邮政编码 (针对模式 2, 3)
	模式 2: (美国) 5 位 + 4 位数
	模式 3: (加拿大) 6 个字母编码, 包括双引号
Lm	符号长度 (忽略双引号), $1 \leq m \leq 138$ , (此参数仅适用于模式 4 和 5)
Message	条码内容

### 示例

SIZE 4,2

GAP 0,0

CLS

SPEED 4

DENSITY 8

OFFSET 0.00

REFERENCE 0,0

SET CUTTER OFF

SET TEAR ON

REM \*\*\*\*\*Mode 2 For USA\*\*\*\*\*

```

MAXICODE 110,100,2,300,840,06810,7317,"DEMO 2 FOR USA MAXICODE"
TEXT 100,520,"3",0,2,2,"Mode 2 For USA"
PRINT 1,1
REM *****Mode 3 For Canada*****
CLS
MAXICODE 110,100,3,300,863,"107317","DEMO 3 FOR CANADA MAXICODE"
TEXT 100,520,"3",0,2,2,"Mode 3 For CANADA"
PRINT 1,1
REM *****MODE4*****
CLS
MAXICODE 110,100,4,"DEMO 4 FOR MAXICODE"
MAXICODE 600,100,4,19,DEMO 4 FOR MAXICODE
TEXT 100,520,"3",0,2,2,"Mode 4 FOR MAXICODE"
PRINT 1,1
REM *****MODE 5*****
CLS
MAXICODE 110,100,5,"DEMO 5 FOR MAXICODE"
MAXICODE 600,100,5,19,DEMO 5 FOR MAXICODE
TEXT 100,520,"3",0,2,2,"DEMO 5 FOR MAXICODE"
PRINT 1

```

## PDF417

该指令用于定义 PDF417 二维码。

### 指令语法

PDF417 x, y, width, height, rotate, [option], expression

参数	描述
x	起点的 X 坐标（以点 dots 表示）
y	起点的 Y 坐标（以点 dots 表示）
width	预期宽度（以点 dots 表示）
height	预期高度（以点 dots 表示）
rotate	逆时针旋转 0: 无旋转 90: 90 度 180: 180 度 270: 270 度
expression	要打印的条码文本或串符号
[option]	
P	数据压缩方式 0: 自动解码



	1: 二进制模式
E	纠错等级。范围: 0~8
M	条码区域的中心图案
	0: 该图案将打印在左上角的区域
	1: 该图案打印在中间区域
Ux, y, c	人工可读
	x: 指定的 x 坐标上人工可读的字符
	y: 指定的 y 坐标上人工可读的字符
	c: 每条线人工可读的最大字符
W	模块宽度 (以点 dots 表示) 范围: 2~9
H	条码高度 (以点 dots 表示) 范围: 4~99
R	最大行数
C	最大列数
T	截断
	0: 不截断
	1: 截断
Lm	符号长度 (不包括双引号), $1 \leq m \leq 2048$

#### 示例

```

SIZE 3,3
GAP 0.12,0
CLS
SPEED 6
DENSITY 8
DIRECTION 1
REFERENCE 0,0
REM *****WITHOUR OPTIONS*****
CLS
PDF417 50,50,400,200,0,"Without Options"
PRINT 1,1
REM *****OPTION:E4*****
CLS
PDF417 50,50,400,200,0,E4,"Error correction level:4"
PRINT 1,1
REM *****OPTION:E4 W4*****
CLS
PDF417 50,50,600,600,0,E4,W4,"Error correction level:4
module width 4 dots"
PRINT 1,1
REM *****OPTION:E4 W4 H4*****
CLS
PDF417 50,50,600,600,0,E4,W4,H4,"Error correction level:4
module width 4 dots

```

```

bar height 4 dots"
PRINT 1,1
REM *****OPTION:E4 W4 H4 R25*****
CLS
PDF417 50,50,600,600,0,E4,W4,H4,R25,"Error correction level:4
Module Width 4 dots
Bar Height 4 dots
Maximum Number of Rows: 25 Rows
"

PRINT 1,1
REM *****OPTION:E4 W4 H4 R40 C3*****
CLS
PDF417 50,50,600,600,0,E4,W4,H4,R40,C3,"Error correction level:4
Module Width 4 dots
Bar Height 4 dots
Maximum Number of Rows: 40 Rows
Maximum number of columns: 3 Cols
"

PRINT 1,1
REM *****OPTION:E4 W4 H4 R40 C4 T0*****
CLS
PDF417 50,50,600,600,0,E4,W4,H4,R40,C4,T0,"Error correction level:4
Module Width 4 dots
Bar Height 4 dots
Maximum Number of Rows: 40 Rows
Maximum number of columns: 4 Cols
Truncation:0
"

PRINT 1,1
REM *****OPTION:E4 W4 H4 R40 C4 T1*****
CLS
PDF417 50,50,900,900,0,E4,W4,H4,R40,C4,T1,"Error correction level:4
Module Width 4 dots
Bar Height 4 dots
Maximum Number of Rows:5 Rows
Maximum number of columns:90 Cols
Truncation:1
"

PRINT 1,1
REM *****OPTION:E4 W4 H4 R40 C4 T0 L169*****
CLS
PDF417 50,50,900,900,0,E4,W4,H4,R40,C4,T0,L169>Error correction level:4
Module Width 4 dots

```

---

```

Bar Height 4 dots
Maximum Number of Rows: 40 Rows
Maximum number of columns: 4 Cols
Truncation:0
Expression length:167
PRINT 1,1
REM *****OPTION:E4 W4 H4 R40 C4 T1 L169*****
CLS
PDF417 50,50,900,900,0,E4,W4,H4,R40,C4,T1,L169,Error correction level:4
Module Width 4 dots
Bar Height 4 dots
Maximum Number of Rows: 40 Rows
Maximum number of columns: 4 Cols
Truncation:1
Expression length:169
PRINT 1,1
REM *****OPTION:P0 E4 W4 H4 R40 C4 T1 L169*****
CLS
PDF417 50,50,900,900,0,P0,E4,W4,H4,R40,C4,T1,L169,Error correction level:4
Module Width 4 dots
Bar Height 4 dots
Maximum Number of Rows: 40 Rows
Maximum number of columns: 4 Cols
Truncation:1
Expression length:169
PRINT 1,1
REM *****OPTION:P0 E4 M0 W6 H6 R60 C4 T0 L283*****
CLS
PDF417 50,50,900,600,0,P0,E4,M0,W6,H6,R60,C4,T0,L283, Data compression method:
P0
Error correction level: E4
Center pattern in barcode area: M0
Human Readable: No
Module Width 6 dots: W6
Bar Height 6 dots: H6
Maximum Number of Rows: 60 Rows: R60
Maximum number of columns: 4 Cols: C4
Truncation:0: T0
Expression length:283: L283
PRINT 1,1
REM *****OPTION:P1 E4 M1 U100,500,10 W4 H4 R60 C4 T1 L297*****
CLS
PDF417 50,50,900,600,0,P1,E4,M1,U100,500,10,W6,H6,R60,C4,T1,L297, Data

```

---

compression method: P1  
 Error correction level: E4  
 Center pattern in barcode area: M1  
 Human Readable: Yes: U100,300,10  
 Module Width 6 dots: W6  
 Bar Height 6 dots: H6  
 Maximum Number of Rows: 60 Rows: R60  
 Maximum number of columns: 4 Cols: C4  
 Truncation:1: T1  
 Expression length:297: L297  
 PRINT 1,1

## AZTEC

该指令用于定义 AZTEC 二维码。

### 指令语法

AZTEC x,y,rotate,[size,]ecp,[flg,]menu,[multi,]rev,]"content"

参数	描述
x	横向起点位置（以点 dots 表示）
y	纵向起点位置（以点 dots 表示）
rotate	旋转 0: 无旋转 90: 90 度 180: 180 度 270: 270 度
size	元素模块大小（1 到 20），默认为 6
ecp	错误控制（&符号大小/类别）参数 0: 默认纠错等级 1~99: 最小的纠错比例 101~104: 1 到 4 阶紧凑符号 201~232: 1 到 32 阶全等符号 300: 一个简单的 Aztec” Rune”
flg	0: 输入信息为直字节 1: 输入用 “<Esc>n” 表示 “FLG(n)”，用 “<Esc> <Esc>” 表示 “<Esc>”
menu	菜单符号（0: 无，1: 有），默认为 0
multi	符号数（1 到 26），默认为 6
rev	相反的输出（0: 无，1: 有），默认为 0
content	条码内容

## 示例

```

SIZE 3,3
GAP 0,0
SPEED 4
DENSITY 8
DIRECTION 0
REFERENCE 0,0
OFFSET 0.00
SET CUTTER OFF
SET TEAR ON
CLS
AZTEC 200,10,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
PRINT 1
CLS
AZTEC 200,10,0,4,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
PRINT 1
CLS
AZTEC 200,10,0,4,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
PRINT 1
CLS
AZTEC 200,10,0,4,1,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
PRINT 1
CLS
AZTEC 200,10,0,4,1,0,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
PRINT 1
CLS
AZTEC 200,10,0,4,1,0,0,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
PRINT 1
CLS
AZTEC 200,10,0,4,1,0,0,1,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
PRINT 1

```

## MPDF417

### Description

This command defines a Micro PDF 417 bar code.

### Syntax

MPDF417 x, y, rotate, [Wn, ][Hn, ][Cn, ]" content"

## PUTBMP

该指令用于打印 BMP 格式的图片。

### 指令语法

PUTBMP X, Y, "filename"

### 参数

### 描述

X	BMP 格式图片的 x 坐标
Y	BMP 格式图片的 y 坐标
filename	下载的 BMP 文件名

### 示例

```
C:\BMP-PCX>DIR
Volume in drive C is WIN98
Volume Serial Number is 4140-4735

Directory of C:\BMP-PCX

01/03/2005  01:06 PM    <DIR>          .
01/03/2005  01:06 PM    <DIR>          ..
01/03/2005  01:52 PM                12,430 TSC.bmp
01/03/2005  01:10 PM                1,181 TSC.pcx
                2 File(s)                13,611 bytes
                2 Dir(s)   8,802,189,312 bytes free

C:\BMP-PCX>COPY CON LPT1
DOWNLOAD "TSC.BMP",12430,^Z
                1 file(s) copied.

C:\BMP-PCX>COPY TSC.BMP/B LPT1
                1 file(s) copied.

C:\BMP-PCX>COPY CON LPT1
SIZE 4,2.5
GAP 0,0
CLS
PUTBMP 100,100,"TSC.BMP"
PRINT 1,1
^Z
                1 file(s) copied.

C:\BMP-PCX>_
```

## PUTPCX

该指令用于打印 PCX 格式的图片。

### 指令语法

PUTPCX X, Y, "filename"

### 参数 描述

X	PCX 图片的 x 坐标
Y	PCX 图片的 y 坐标
filename	下载的 PCX 文件名。区分大小写。

### 示例

```
C:\BMP-PCX>DIR
Volume in drive C is WIN98
Volume Serial Number is 4140-4735

Directory of C:\BMP-PCX

01/03/2005  01:06 PM    <DIR>          .
01/03/2005  01:06 PM    <DIR>          ..
01/03/2005  01:52 PM                12,430 TSC.bmp
01/03/2005  01:10 PM                1,181 TSC.pcx
                2 File(s)                13,611 bytes
                2 Dir(s)      8,802,189,312 bytes free

C:\BMP-PCX>COPY CON LPT1
DOWNLOAD "TSC.PCX",1181,^Z
                1 file(s) copied.

C:\BMP-PCX>COPY TSC.PCX/B LPT1
                1 file(s) copied.

C:\BMP-PCX>COPY CON LPT1
SIZE 4,2.5
GAP 0,0
CLS
PUTPCX 100,100,"TSC.PCX"
PRINT 1,1
^Z
                1 file(s) copied.

C:\BMP-PCX>_
```

## QR CODE

该指令用于打印二维码。

### 指令语法

QR CODE X, Y, ECC Level, cell width, mode, rotation, [model, mask,]"Data string"

参数	描述
X	二维码的左上角 x 坐标
Y	二维码的左上角 y 坐标
ECC level	纠错恢复等级
	L: 70% L: 7%
	M: 15%
	Q: 25%
	H: 30%
条码宽度	1~10
模式	自动/手动解码
	A: 自动
	M: 手动
旋转	0: 0 度
	90: 90 度
	180: 180 度
	270: 270 度
型号	M1: (默认), 原始版本
	M2: 加强版本
掩码	S0~S8, 默认 S7
数据串	可编码字符集如下所示

可编码字符集:

- 1) 数字数据: (位数 0~9)
- 2) 文字数字数据  
位数 0~9  
大写字母 A-Z  
其余 9 个字符: 空格, \$ % \* + - . / : ;
- 3) 8 比特数据  
JIS 8 比特字符集 (拉丁语和日语) 与 JIS X 0201 相符
- 4) kanji 字符  
转换 JIS 值 8140<sub>HEX</sub> - 9FFC<sub>HEX</sub> and E040<sub>HEX</sub> - EAA4<sub>HEX</sub>。这些是从 JIS X 0208 转换的值。

每个符号的数据字符 (对于最大的符号尺寸来说):

	型号 1 (版本 14-L)	型号 2 (版本 40-L)
1) 数字数据:	1,167 个字符	7,089 个字符
2) 文字数字数据:	707 个字符	4,296 个字符



- 3) 8 比特数据: 486 个字符                      2,953 个字符  
4) kanji 数据: 299 个字符                      1,817 个字符

\*如果“A”是数据字符串中的第一个字符，那么“A”后面的数据是文字数字数据。  
\*如果“N”是数据字符串中的第一个字符，那么“N”后面的数据是数字数据。  
\*如果“B”是数据字符串中的第一个字符，那么“B”后面的 4 位数字是用于指定数据的数量。  
在这 4 位数之后是待编码的二进制数据的字节数。  
\*如果“K”是数据字符串中的第一个字符，那么“K”后面的数据就是 Kanji 数据。  
\*如果“!”在数据字符串中，后面跟着“N”、“A”、“B”、“K”，然后它将被切换到指定的可编码字符集。

## 示例

### 手动模式:

QRCODE 100,10,L,7,M,0,M1,S1,"ATHE FIRMWARE HAS BEEN UPDATED"  
(其中 A: 文字数字数据)

QRCODE 100,10,M,7,M,0,M1,S2,"N123456"  
(其中 N: 数字数据)

QRCODE 100,10,Q,7,M,0,M1,S3,"N123456!ATHE FIRMWARE HAS BEEN  
UPDATED"  
(其中 N: 数字数据; ! : 转换字符; A: 文字数字数据)

QRCODE 100,10,H,7,M,0,M1,S3,"B0012Product name"  
(其中 B: 二进制数据; 0012: 12 比特)

QRCODE 100,10,M,7,M,0,M1,S3,"K"  
(其中 K: Kanji 数据)

### 自动模式:

QRCODE 100,10,M,7,A,0,"THE FIRMWARE HAS BEEN UPDATED"

(1) 自动模式示例

#### a. General data string

SIZE 4,2.5

GAP 0.12,0

CLS

QRCODE 10,10,H,4,A,0,"ABCabc123"

QRCODE 160,160,H,4,A,0,"123ABCabc"

QRCODE 310,310,H,4,A,0,"印表機 ABCabc123"

PRINT 1,1

**b. Data string including character (0Dh, 0Ah)**

SIZE 4,2.5

GAP 0.12,0

CLS

QRCODE 10,10,H,4,A,0,"ABC

abc

123"

QRCODE 160,160,H,4,A,0,"123

ABC

abc"

QRCODE 310,310,H,4,A,0,"印表機

ABC

abc

123"

PRINT 1,1

**c. Data string concatenation (Must be used with DOWNLOAD ... EOP command)**

DOWNLOAD "DEMO.BAS"

SIZE 4,2.5

CAP 0.12,0

CLS

QRCODE 10,10,H,4,A,0,"ABCabc123"+STR\$(1234)

QRCODE 160,160,H,4,A,0,"123ABCabc"+"1234"

QRCODE 310,310,H,4,A,0,"印表機 ABCabc123"+"1234"+"abcd"

PRINT 1,1

EOP

DEMO

**d. Data string including double quote ( " ) character, please use \["] instead of**

SIZE 4,2.5

CAP 0.12,0

CLS

QRCODE 10,10,H,4,A,0,"ABC\["]abc\["]123"

QRCODE 160,160,H,4,A,0,"123\["]ABC\["]abc"

QRCODE 310,310,H,4,A,0,"\["]印表機\["]ABCabc123"

PRINT 1,1

**(3) Manual mode**

**a. General data string:**

SIZE 4,2.5

CAP 0.12,0

CLS

QRCODE 10,10,H,4,M,0,"AABC!B0003abc!N123"

QRCODE 160,160,H,4,M,0,"N123!AABC!B0003abc"

QRCODE 310,310,H,4,M,0,"K 印表機!AABC!B0006abc123"

PRINT 1,1

**b. Data string including character, is an 8-bit byte data**

SIZE 4,2.5

CAP 0.12,0

CLS

QRCODE 10,10,H,4,M,0,"AABC!B0007

abc

!N123"

QRCODE 160,160,H,4,M,0,"N123!B0002

!AABC!B0005

abc"

QRCODE 310,310,H,4,M,0,"K 印表機!B0002

!AABC!B0010

abc

123"

PRINT 1,1

**c. Data string concatenation (Must be used with DOWNLOAD ... EOP command)**

DOWNLOAD "A.BAS"

SIZE 4,2.5

CAP 0.12,0

CLS

QRCODE 10,10,H,4,M,0,"AABC!B0006abc123!N"+STR\$(1234)QRCODE

160,160,H,4,M,0,"N123!AABC!B0007abc"+"1234"

QRCODE 310,310,H,4,M,0,"K 印表機!AABC!B0014abc123"+"1234"+"abcd"

PRINT 1,1

EOP

A

**d. Data string including double quote ( " ) character, please use \" instead of**

SIZE 4,2.5

CAP 0.12,0

CLS

QRCODE 10,10,H,4,M,0,"AABC!B0005\"abc\"!N123"

QRCODE 160,160,H,4,M,0,"N123!B0001\"!AABC!B0004\"abc"

QRCODE 310,310,H,4,M,0,"B0001\"!K 印表機!B0010\"ABCabc123"

PRINT 1,1

## RSS

该指令用于在标签上制作一个 RSS 条码。

### 指令语法

RSS x,y,"sym",rotate,pixMult,sepHt,"content"

RSS x,y,"RSSEXP",rotate,pixMult,sepHt,segWidth,"content"

RSS x,y,"UCC128CCA",rotate,pixMult,sepHt,linHeight,"content"

RSS x,y,"UCC128CCC",rotate,pixMult,sepHt,linHeight,"content"

参数	描述
x	x 坐标
y	y 坐标
sym	符号类型： RSS14: RSS14 RSS14T: RSS14 Truncated RSS14S: RSS14 Stacked RSS14SO: RSS14 Stacked Omnidirectional RSSLIM: RSS Limited RSSEXP: RSS Expanded UPCA: UPC-A UPCE: UPC-E EAN13: EAN-13 EAN8: EAN-8 UCC128CCA: UCC/EAN-128 & CC-A/B UCC128CCC - UCC/EAN-128 & CC-C
Rotate	旋转（0,90,180,和 270 有效）
pixMult	每 X 的像素（1 到 10 有效）
sepHt	分隔行的高度（1 和 2 有效）
segWidth	RSS 扩展的段落宽度（2 到 22 有效）
linHeight	X 里的 UCC/EAN-128 高度（1 到 500 有效）
content	条码内容或串符号

备注: **200 DPI: 1 mm = 8 dots**  
**300 DPI: 1 mm = 12 dots**

反黑区域推荐的最大高度为 12mm，宽度为 4 英寸。高度大于 12 毫米的反向区域可能损坏电源，影响打印质量。

最大印字率因机型而异。台式打印机和工业打印机的印字率的上限分别为 20% 和 30%。

## 示例

```
SIZE 100 mm,100 mm
GAP 0,0
CLS
RSS 300,300,"RSS14",0,2,2,"1234567890|ABCDEFGF"
RSS 300,300,"RSS14T",90,2,2,"1234567890|ABCDEFGF"
RSS 300,300,"RSS14S",180,2,2,"1234567890|ABCDEFGF"
RSS 300,300,"RSS14SO",270,2,2,"1234567890|ABCDEFGF"
PRINT 1,1
SIZE 100 mm,100 mm
GAP 0,0
CLS
RSS 300,300,"RSSLIM",0,2,2,"1234567890|ABCDEFGF"
RSS 300,300,"RSSEXP",90,2,2,22,"1234567890|ABCDEFGF"
RSS 300,300,"UPCA",180,2,2,"1234567890|ABCDEFGF"
RSS 300,300,"UPCE",270,2,2,"000|ABCDEFGF"
PRINT 1,1
SIZE 100 mm,100 mm
GAP 0,0
CLS
RSS 300,300,"EAN13",0,2,2,"123456789012|ABCDEFGF"
RSS 300,300,"EAN8",90,2,2,"1234567|ABCDEFGF"
RSS 300,300,"UCC128CCA",180,2,2,25,"1234567890|ABCDEFGF"
RSS 300,300,"UCC128CCC",270,2,2,25,"1234567890|ABCDEFGF"
PRINT 1,1
```

## REVERSE

该指令用于图片缓冲区的反向打印。

### 指令语法

REVERSE X\_start, Y\_start, X\_width, Y\_height

参数	描述
X_start	起点的 x 坐标
Y_start	起点的 y 坐标
X_width	x 轴区域的宽度（以点 dots 表示）
Y_height	y 轴区域的高度（以点 dots 表示）

备注: **200 DPI: 1 mm = 8 dots**  
**300 DPI: 1 mm = 12 dots**

反黑区域推荐的最大高度为 12mm，宽度为 4 英寸。高度大于 12 毫米的反向区域可能损坏电源，影响打印质量。

最大印字率因机型而异。台式打印机和工业打印机的印字率分别限制在 20% 和 30%。

## 示例

SIZE 4,2.5

GAP 0,0

SPEED 6

DENSITY 8

DIRECTION 0

CLS

TEXT 100,100,"3",0,1,1,"REVERSE"

REVERSE 90,90,128,40

PRINT 1,1

## TEXT

该指令用于在标签上打印文本。

### 指令语法

TEXT X, Y, "font", rotation, x-multiplication, y-multiplication, "content"

#### 参数

#### 描述

X 文本的 x 坐标

Y 文本的 y 坐标

font 字体名称

0: 单色 CG Triumvirate 加粗浓缩，字体宽度和高度可变

1: 8×12 固定点数字体

2: 12×20 固定点数字体

3: 16×24 固定点数字体

4: 24×32 固定点数字体

5: 32×48 固定点数字体

6: 14×19 固定点数字体

7: 21×27 固定点数字体

8: 14×25 固定点数字体

ROMAN.TTF: 单色 CG Triumvirate 加粗浓缩，字体宽度和高度固定

rotation 文本旋转角度

alignment :  
0 : Default (Left)  
1 : Left  
2 : Center  
3 : Right

0: 无旋转  
90: 90 度, 顺时针  
180: 180 度, 顺时针  
270: 270 度, 顺时针

x-multiplication: 横向放大倍率, 最大 10x  
有效范围: 1~10

对于"ROMAN.TTF" 真实字体, 请忽略此参数。

对于字体"0", 该参数用于指定真实的字体宽度 (point)。1 point=1/72 inch.

y-multiplication: 纵向放大倍率, 最大 10x  
有效范围: 1~10

对于 *Truetype* 字体, 此参数用于指定其高度 (point)。1 point = 1/72 inch

### 示例

SIZE 3,2

GAP 0,0

CLS

TEXT 100,100,"5",0,1,1,"["]DEMO FOR TEXT\["]"

TEXT 100,200,"ROMAN.TTF",0,1,20,"["]True Type Font Test Print\["]"

PRINT 1,1

### \*BLOCK

#### Description

This command prints paragraph on label.

#### Syntax

BLOCK x,y,width,height," font" ,rotation,x-multiplication,y-multiplication,[space,]  
alignment,]  
" content"

## 3. 询问打印机状态指令（RS-232）

### <ESC>!?

询问打印机状态指令为立即回应型指令，该指令控制字元是以<ESC>（ASCII 27, escape 字元）为控制字元，即使打印机在错误状态中仍能通过 RS-232 回传一个 byte 字符来表示打印机状态，若回传值为 0 则表示打印机处于正常的状态。

#### 指令语法

<ESC>!?

Bit（回传值）	状态
0	打印头未到位
1	卡纸
2	缺纸
3	缺碳带
4	暂停
5	正在打印
6	开盖（可选）

十六进制	打印机状态
00	正常
01	打印头未到位
02	卡纸
03	卡纸且打印头未到位
04	缺纸
05	缺纸且打印头未到位
08	缺碳带
09	缺碳带且打印头未到位
0A	缺碳带且卡纸
0B	缺碳带，卡纸且打印头未到位
0C	缺碳带且缺纸
0D	缺碳带，缺纸且打印头未到位
10	暂停
20	正在打印
80	其他错误



## <ESC>!R

该指令用于重置打印机。指令开头是一个 ESCAPE 字符（ASCII 27）。下载到存储器的文件会被删除。不能在转储模式下发送这条指令。

## ~!@

该指令用于询问打印机的里程，回传值以 ASCII 字符表示，只回传里程的整数部分（小数点部分不回传），以 0x0D 结束。

## ~!A

该指令用于询问打印机的内存大小。回传值以十进制字元表示，以 0x0d 结束。

## ~!C

该指令用于询问是否安装了实时时钟。由打印机发出一个字节，指示是否装有 RTC。

回传值	描述
0	未安装 RTC
1	已安装 RTC

## ~!D

该指令用于使打印机进入转储模式。在转储模式下，打印机不转译直接输出代码。

## ~!F

该指令用于询问打印机存储器所有的文件以及所安装的字体。文件名的返回值为 ASCII 字符，每个文件名以 0x0D 结束。结束的字符为 0x1A，多次输入这条指令将会循环存储器的文件。

## ~!I

该指令用于询问打印机的代码页和国家代码设置。

返回的信息格式如下：

代码页，国家代码

ex: 8 bit: 437,001

7 bit: USA, 001

关于打印机支持的代码页和国家代码，请分别参照 CODEPAGE 和 COUNTRY 的指令。

## ~!T

该指令用于询问打印机的型号。回传信息以 ASCII 字符表示。

## 4. 文件管理指令

### DOWNLOAD

“DOWNLOAD”是要保存在打印机内存中的文件的标头。下载的文件可分为两类：程序文件和数据文件(包括文本数据文件，PCX 图形文件和位图字体文件)。

#### 指令语法

##### 1. 下载程序文件：

DOWNLOAD [n,] “FILENAME.BAS”

#### 参数

#### 描述

n

指定用于保存下载文件的存储器

**忽略 n:** 仅下载文件到 DRAM。如果你想在关机之前将文件从 DRAM 保存到闪存里，发送 MOVE 指令给打印机。

**F:** 下载文件到主板闪存。

**E:** 下载文件到扩展内存单元。

FILENAME.BAS 存储在打印机内存里的文件名。

#### 备注：

- 1) 文件名要区分大小写。
  - 2) 文件扩展必须是“.BAS”。
  - 3) 文件名必须是 8.3 格式。
  - 4) 必须跟 EOP 指令一起使用。
  - 5) 如果没有指定存储器，所有文件都会下载到 DRAM。
- 电池没电的情况下，DRAM 备份的文件会丢失。

##### 2. 下载数据文件：

DOWNLOAD [n,] "FILENAME", DATA SIZE, DATA CONTENT...

#### 参数

#### 描述

指定用于保存下载文件的存储器

**忽略 n:** 仅下载文件到 DRAM。如果你想在关机之前将文件从 DRAM 保存到闪存里，发送 MOVE 指令给打印机。

**F:** 下载文件到主板闪存。

**E:** 下载文件到扩展内存单元。

FILENAME.BAS 将会继续存储在打印机内存里的文件名（区分大小写）。

DATA SIZE 数据文件的实际大小（以 bytes 表示，不含标头）

DATA CONTENT 将要下载到打印机的数据

#### 备注：

- 1) 对于文本数据文件，CR（回车）0×0D 和 LF（换行）0×0A 为数据的分隔符。
  - 2) 如果没有指定存储器，所有文件都会下载到 DRAM
- 电池没电的情况下，DRAM 备份的文件会丢失。

## 示例

以下示例的程序将会下载到打印机 SDRAM。

```
DOWNLOAD "EXAMPLE.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 100,100,"3",0,1,1,"EXAMPLE PROGRAM"
PRINT 1
EOP
```

备注：当下载程序时，“**DOWNLOAD**”标头必须放在文件之首，“**EOP**”必须放在程序之后。要运行程序时，不需要 **BAS** 扩展回调主文件名即可，或使用 **RUN** 这条指令来运行下载程序。

示例：

回调主文件名

```
C:\>COPY CON LPT1<ENTER>
EXAMPLE<ENTER>
<CTRL><Z>
C:\>
```

使用 **RUN** 指令运行程序

```
C:\>COPY CON LPT1<ENTER>
RUN "EXAMPLE.BAS"<ENTER>
<CTRL><Z>
C:\>
```

以下是下载数据文件的示例

```
DOWNLOAD "DATA",20,COMPUTER<Enter>
2001<Enter>
21<Enter>
```

备注：<ENTER>表示键盘上的“ENTER”键。在以上的示例中，请按下“ENTER”键，而不是输入<ENTER>。

## EOP

该指令用于结束程序。为了表示程序中使用的 BASIC 语言指令的开头和结尾，DOWNLOAD "FILENAME.BAS "必须加在程序的首行，“EOP”放在程序的末行。

### 示例

```
DOWNLOAD "DEMO.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 100,100,"3",0,1,1,"DEMO PROGRAM"
PRINT 1
EOP
```

## FILES

该指令用于打印存储器大小、可用存储大小及打印机存储器（包括 FLASH 和 DRAM）里的文件列表（或通过 RS-232 的文件列表）。

### 示例

通过串口或并口连接在 DOS 环境下，按照以下的步骤打印出保存在打印机存储器里的文件列表。

```
C:\>MODE COM1 96,N,8,1<ENTER>
C:\>COPY CON COM1<ENTER>
FILES<ENTER>
<CTRL><Z><ENTER>
C:\>
或者:
C:\>COPY CON LPT1<ENTER>
FILES<ENTER>
<CTRL><Z><ENTER>
```

**备注：**<ENTER>表示键盘上的“ENTER”键。<CTRL><Z>表示同时按住键盘上的“CTRL”键和<Z>键。

## KILL

该指令用于删除打印机存储器的文件。外卡将删除 DRAM 和 FLASH 存储器里所有的文件。

### 指令语法

KILL [n], "FILENAME"

### 参数

### 描述

- n 指定要删除的文件的存储位置。
- 忽略 n:** 删除保存在 DRAM 的文件。
- F:** 删除主板闪存里的文件。
- E:** 删除扩展存储里的文件。

### 备注:

1) 如果未指定可选参数 n，固件将会删除 DRAM 的文件。

2) 语法示例

**KILL "FILENAME ":** 删除 DRAM 的指定文件

**KILL "\*.PCX ":** 删除 DRAM 所有的 PCX 文件

**KILL "\*" :** 删除 DRAM 所有的文件

**KILL F, "FILENAME ":** 删除 FLASH 的指定文件

**KILL E, "\*.PCX ":** 删除扩展存储卡的所有 PCX 文件

### 示例

用户可以使用打印机 SELFTEST 工具列出保存在打印机存储器里的打印机配置和文件，或者使用 FILES 指令打印下载文件。参照以下步骤，通过并口连接来删除打印机存储器里的文件。

```
C:\>COPY CON LPT1<ENTER>
```

```
FILES<ENTER>
```

```
<CTRL><Z><ENTER>
```

```
C:\>COPY CON LPT1<ENTER>
```

```
KILL "DEMO.BAS " <ENTER>
```

```
<CTRL><Z><ENTER>
```

```
C:\>COPY CON LPT1<ENTER>
```

```
FILES<ENTER>
```

```
<CTRL><Z><ENTER>
```

**备注:** <ENTER>表示键盘上的“ENTER”键。<CTRL><Z>表示同时按住键盘上的“CTRL”键和<Z>键。

## MOVE

该指令将 DRAM 的下载文件移至 FLASH 存储器。

## RUN

该指令执行打印机存储器里的程序。

### 指令语法

RUN "FILENAME.BAS"

### 示例

C:\>COPY CON LPT1<ENTER>

RUN "DEMO.BAS "<ENTER>

<CTRL><Z><ENTER>

C:\>

### 具体示例：

DOWNLOAD " DEMO.BAS"

SIZE 4,4

GAP 0,0

DIRECTION 1

SET TEAR ON

CLS

TEXT 100,100," 3" ,0,1,1," DEMO PROGRAM"

PRINT 1

EOP

RUN " DEMO.BAS"

备注：<ENTER>表示键盘上的“ENTER”键。<CTRL><Z>表示同时按住键盘上的“CTRL”键和<Z>键。

## 5. BASIC 指令及函数

### ABS()

此函数回传整数、浮点或变量的绝对值。

#### 指令语法

ABS (VARIABLE)

#### 示例

```
DOWNLOAD "TEST.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=ABS(-100)
B=ABS(-50.98)
C=-99.99
TEXT 100,100,"3",0,1,1,STR$(A)
TEXT 100,150,"3",0,1,1,STR$(B)
TEXT 100,200,"3",0,1,1,STR$(ABS(C))
PRINT 1
EOP
RUN " TEST.BAS"
```

### ABC()

此函数回传字符的 ASCII 码。

#### 指令语法

ASC ("A")

#### 示例

```
DOWNLOAD "TEST.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
```

```
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
CODE1=ASC("A ")
TEXT 100,100, "3",0,1,1,STR$(CODE1)
PRINT 1
EOP
```

## CHR\$( )

此函数回传指定的 ASCII 码的字符。

### 指令语法

CHR\$(n)

参数	描述
n	ASCII 码

### 示例

```
DOWNLOAD "TEST.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=65
WORD$=CHR$(A)
TEXT 100,100, "3",0,1,1,WORD$
PRINT 1
EOP
```



## END

该指令指示程序的结束点。

### 示例

```

DOWNLOAD "DEMO.BAS"
SIZE 4,2
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 200,60,"4",0,1,1,"END COMMAND TEST"
X=300
Y=200
X1=500
Y1=400
GOSUB DR_LINE
PRINT 1
END

:DR_LINE
FOR I=1 TO 100 STEP 10
BOX X+I,Y+I,X1-I,Y1-I,5
NEXT
RETURN
EOP
DEMO

```

## EOF()

此函数用于判断打开的下载文件是否已到文件的结束位置。

### 指令语法

EOF (File Handle)

参数	描述
文件句柄	0 或 1
返回值	描述
无-0	文档结束点
0	非文档结束点

### 示例

```
DOWNLOAD "DATA",16,COMPUTER
2000
```

```
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.0,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA",0
SEEK 0,0
Y=110
TEXT 10,10,"3",0,1,1,"*****EOF TEST*****"
:A
Temp$=""
READ 0,ITEM$,P
TEXT 10,Y,"2",0,1,1,ITEM$+"$"+STR$(P)+"[EOF(0)="+STR$(EOF(0))+"]"
BARCODE 10,Y+25,"39",40,1,0,2,4,"PRICE-"+STR$(P)
Y=Y+100
IF EOF(0)=0 THEN GOTO A
PRINT 1
EOP
DEMO
```

## OPEN

该指令用于打开下载的文件，创建文件句柄。最多支持两个文件句柄，因此最多只能同时打开两个文件。若要打开文件，必须在使用这条指令之前下载该文件。

当打开一个文件时，固件会自动搜索文件，判断该文件是在主板的闪存里还是在扩展存储卡里。若没有此文件，打印机将会创建一个文件。

### 指令语法

OPEN "Filename", File handle OPEN [memory ID,]" filename" ,file handle

### 参数

### 描述

文件名 下载到打印机存储器的文件  
文件句柄 0 或 1

### 示例

如果要下载一个以“DATA”命名的文件，该文件的格式要包含：

DOWNLOAD "DATA1",56,COMPUTER

2000

12

MOUSE

500

13

KEYBOARD

300

100

DOWNLOAD "DATA2",56,Computer

3000

32

Mouse

900

93

Keyboard

700

700

示例：

DOWNLOAD " DATA.DAT" ,18,Open file in DRAM.

DOWNLOAD F," DATA.DAT" ,19,Open file in FLASH.

DOWNLOAD " TEST.BAS"

data1\$=" "

data2\$=" "

data3\$=" "

OPEN " DATA.DAT" ,0

READ 0,data1\$

CLOSE 0

OPEN F," DATA.DAT" ,0

READ 0,data2\$

CLOSE 0

KILL F," \*"

OPEN " NEW.DAT" ,0

SEEK 0,0

WRITE 0," Auto create a new file in FLASH."

SEEK 0,0

READ 0,data3\$

CLOSE 0

SIZE 4,1

GAP 0,0

CLS

TEXT 10,10," 3" ,0,1,1,data1\$

TEXT 10,60," 3" ,0,1,1,data2\$

TEXT 10,110," 3" ,0,1,1,data3\$

PRINT 1

EOP

TEST

将上述数据内容保存在“DATA”的文件名下。按下列步骤将数据下载到打印机。

C:\>COPY DATA/B LPT1

如果要下载一个以” DEMO.BAS”命名的文件，该文件的格式要包含：

```

DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 4
DIRECTION 0
I=1
Y=100
GOSUB OpenData
:Start
CLS
TEXT 10,10,"3",0,1,1,"*****OPEN COMMAND TEST*****"
ITEM$=""
READ 0,ITEM$,P,Q
TEXT 10,Y,"2",0,1,1,ITEM$+"$"+STR$(P)+"[EOF(0)="+STR$(EOF(0))+"]"
BARCODE 10,Y+25,"39",40,1,0,2,4,"PRICE*"+STR$(Q)+"="+STR$(P*Q)
Y=Y+100
PRINT 1
Y=100
IF EOF(0)=1 THEN GOSUB OpenData
IF EOF(0)=0 THEN GOTO Start
END
:OpenData
IF I=1 THEN OPEN "DATA1",0
IF I=2 THEN OPEN "DATA2",0
SEEK 0,0
IF I>2 THEN END
I=I+1
RETURN
EOP
DEMO

```

将上述数据内容保存在” DEMO”的文件名下。按下列步骤将数据下载到打印机。

```
<under MS-DOS mode>:
C:\>COPY DEMO/B LPT1
Execute DEMO.BAS in printer:
C:\>COPY CON LPT1
DEMO
<Ctrl><Z>
```

以上示例指示打印机以同样的文件句柄 0 来打开文件“DATA1”和“DATA2”，读取文件的项目。

## WRITE

该指令用于将数据写入下载的数据文件。可以通过两个文件句柄的打印机同时打开两个文件。

### 指令语法

WRITE file handle, variables

参数	描述
file handle	0 或 1
variables	字符串，整数或浮点变量

## READ

该指令用于读取下载文件的数据。

### 指令语法

READ file handle, variables

参数	描述
file handle	0 或 1
variables	字符串，整数或浮点变量

### 示例

```
DOWNLOAD "DATA1",20,COMPUTER
2000
12
DOWNLOAD "DATA2",16,Mouse
900
93
DOWNLOAD "DEMO.BAS"
SIZE 3,1
```

---

```

GAP 0,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
I=0
Y=100
OPEN "DATA1",0
OPEN "DATA2",1
SEEK 0,0
SEEK 1,0
:Start
CLS
TEXT 10,10,"3",0,1,1,"*****READ COMMAND TEST*****"
TEXT 10,50,"3",0,1,1,"OPEN-READ DATA"+STR$(I+1)
ITEM$=""
READ I,ITEM$,P,Q
TEXT 10,Y,"2",0,1,1,ITEM$+"$"+STR$(P)
BARCODE 10,Y+25,"39",40,1,0,2,4,"PRICE*"+STR$(Q)+"="+STR$(P*Q)
Y=Y+100
PRINT 1
Y=100
IF I<=1 THEN
IF EOF(I)=1 THEN
I=I+1
GOTO Start
ELSE
GOTO Start
ENDIF
ELSE
END
ENDIF
EOP
DEMO

```

---

## SEEK

该指令用于移动文件指针至一个特定位置。

### 指令语法

SEEK file handle, offset

参数	描述
file handle	0 或 1
offset	移至新位置的偏移字符

### 示例

DOWNLOAD "DATA",12,1234567890

DOWNLOAD "TEST.BAS"

SIZE 3,1

GAP 0,0

DENSITY 8

SPEED 3

DIRECTION 1

REFERENCE 0,0

CLS

OPEN "DATA",0

SEEK 0,4

READ 0,Num\$

TEXT 100,10,"3",0,1,1,"SEEK COMMAND TEST"

BAR 100,40,300,4

TEXT 100,60,"3",0,1,1,"SHIFT 4 CHARACTERS"

TEXT 100,110,"3",0,1,1,Num\$

BAR 100,140,300,4

SEEK 0,0

READ 0,Num\$

TEXT 100,160,"3",0,1,1,"SHIFT 0 CHARACTERS"

TEXT 100,210,"3",0,1,1,Num\$

PRINT 1

EOP

TEST

## LOF()

此函数用于回传指定文件的大小。

### 指令语法

LOF ("FILENAME")

### 参数 描述

FILENAME 下载到打印机存储器的文件。

### 示例

```
DOWNLOAD "DATA1",10,1234567890
```

```
DOWNLOAD "DATA2",15,ABCDEFGHJKLMNO
```

```
DOWNLOAD "LofTest.BAS"
```

```
SIZE 3,3
```

```
GAP 0.08,0
```

```
DENSITY 8
```

```
SPEED 3
```

```
DIRECTION 0
```

```
REFERENCE 0,0
```

```
SET CUTTER OFF
```

```
SET PEEL OFF
```

```
CLS
```

```
OPEN "DATA1",0
```

```
OPEN "DATA2",1
```

```
TEXT 10,20,"4",0,1,1,"LOF() FUNCTION TEST"
```

```
J=LOF("DATA1")
```

```
K=LOF("DATA2")
```

```
TEXT 10,140,"3",0,1,1,"DATA1 IS: "+STR$(J)+" Bytes"
```

```
TEXT 10,200,"3",0,1,1,"DATA2 IS: "+STR$(K)+" Bytes"
```

```
PRINT 1
```

```
EOP
```

```
LofTest
```



## FREAD\$( )

此函数用于读取文件的数据字节数。

### 指令语法

FREAD\$(file handle, byte)

参数	描述
file handle	0 或 1
byte	读取的字节数

### 示例

```
DOWNLOAD "DATA1",10,1234567890
```

```
DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO
```

```
DOWNLOAD "OPEN2.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
OPEN "DATA1",0
OPEN "DATA2",1
SEEK 0,0
SEEK 1,0
Y$=FREAD$(0,6)
Z$=FREAD$(1,6)
TEXT 10,260,"3",0,1,1,"FREAD$(0,6) IS: "+Y$
TEXT 10,320,"3",0,1,1,"FREAD$(1,6) IS: "+Z$
PRINT 1
EOP
```

## FOR...NEXT LOOP

该指令用于重复执行单行或多行程序。循环计数器的值指定执行次数。打印机允许嵌套循环（多达 39 个嵌套循环）。禁止从 FOR...NEXT 循环的中间跳出。

### 指令语法

```
FOR variable = start TO end STEP increment
statement; start < end
NEXT
```

参数	描述
变量	变量名称（最多 8 个字符）
起点	整数或浮点数
终点	整数或浮点数
增量	整数或浮点数，正数或负数

### 示例

```
DOWNLOAD "TEST.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 1
CLS
FOR I=1 TO 10 STEP 1
TEXT 100,10+30*(I-1),"3",0,1,1,STR$(I)
NEXT
FOR I=1 TO 1000 STEP 100
TEXT 200,10+((I-1)/10)*3,"3",0,1,1,STR$(I)
NEXT
FOR I=110 TO 10 STEP -10
TEXT 300,10+(ABS(I-110))*3,"3",0,1,1,STR$(I)
NEXT
FOR I=1 TO 5 STEP 0.5
IF I-INT(I)=0 THEN Y=10+60*(I-1) ELSE Y=Y+30
TEXT 400,Y,"3",0,1,1,STR$(I)
NEXT
PRINT 1
EOP
TEST
```

## IF...THEN...ELSE...ENDIF LOOP

该指令用于执行单个或多个条件式语句。可使用单行或多行“区块”语法。

### 指令语法

IF *condition* THEN *statement*

注意单行格式的 IF...THEN 不使用 ENDIF 语句。

参数	描述
condition	可用相关分隔符：<, >, =, <=, >=
statement	仅一个语句可用

### 示例

```
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.12,0
SPEED 4
DENSITY 8
DIRECTION 1
REFERENCE 0,0
OFFSET 0.00
SET CUTTER OFF
SET PEEL OFF
CLS
```

```
A=0
B=0
C=0
D=0
E=0
F=0
G=0
H=0
J=0
K=0
L=0
```

```
FOR I=1 TO 100
  IF I-INT(I/1)*1=0 THEN A=A+I
```

```
IF I-INT(I/2)*2=1 THEN B=B+I ELSE C=C+I
  IF I-INT(I/3)*3=0 THEN
    D=D+I
```

```

ENDIF
IF I-INT(I/5)*5=0 THEN
E=E+I
ELSE
F=F+I
ENDIF
IF I-INT(I/7)*7=0 THEN
G=G+I
ELSEIF I-INT(I/17)*17=0 THEN
H=H+I
ELSEIF I-INT(I/27)*27=0 THEN
J=J+I
ELSEIF I-INT(I/37)*37=0 THEN
K=K+I
ELSE
L=L+I
ENDIF
NEXT
TEXT 100,110,"3",0,1,1,"(1) 1+2+3+...+100="+STR$(A)
TEXT 100,160,"3",0,1,1,"(2) 1+3+5+...+99="+STR$(B)
TEXT 100,210,"3",0,1,1,"(3) 2+4+6+...+100="+STR$(C)
TEXT 100,260,"3",0,1,1,"(4) 3+6+9+...+99="+STR$(D)
TEXT 100,310,"3",0,1,1,"(5) 5+10+15+...+100="+STR$(E)
TEXT 100,360,"3",0,1,1,"(1)-(5)="+STR$(F)
TEXT 100,410,"3",0,1,1,"(6) 7+14+21+...+98="+STR$(G)
TEXT 100,460,"3",0,1,1,"(7) 17+34+51+...+85="+STR$(H)
TEXT 100,510,"3",0,1,1,"(8) 27+54+...+81="+STR$(J)
TEXT 100,560,"3",0,1,1,"(9) 37+74="+STR$(K)
TEXT 100,610,"3",0,1,1,"(1)-(6)-(7)-(8)-(9)="+STR$(L)
PRINT 1,1
EOP
DOWNLOAD "IFTHEN.BAS"
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=50
B=5

```

```
C$=""
D$=""
:L1
IF A>100 THEN GOTO L1 ELSE A=A+10
C$=STR$(A)+" IS SMALLER THAN 100"
TEXT 100,10,"4",0,1,1,C$
PRINT 1
END
:L2
A=A+B
D$=STR$(A)+" IS LARGER THAN 100"
TEXT 100,100,"4",0,1,1,D$
PRINT 1
GOTO L1
EOP
```

**备注:**

如果表达式的结果为非零,则会执行下面的 **THEN** 语句。如果表达式的结果为零,且存在 **ELSE** 后面的语句,则会执行该语句。 否则将执行下一行语句。

如果 **IF...THEN ...ELSE** 中存在语句块,则必须在其末尾使用 **ENDIF**。

**限制:**

程序中嵌套的 **IF.THEN.ELSE** 语句总数不能超过 40。

一个程序中嵌套的 **IF ...THEN ...ELSE, FOR...NEXT, GOSUB RETURN** 总数不能超过 40 个循环。

## GOSUB...RETURN

该指令用于由主程序跳到子程序，在到达“RETURN”之前执行语句。

### 指令语法

```
GOSUB LABEL
    statement
END
:LABEL
    statement
RETURN
```

### 参数

### 描述

LABEL 子程序的起点。标签的最大长度为 8 字符。

### 示例

```
SIZE 3,3
GAP 0,0
DENSITY 8
SPEED 4
DIRECTION 0
CLS
TEXT 10,10,"3",0,1,1,"GOSUB & RETURN COMMAND TEST"
GOSUB DR_BOX
PRINT 1
END
:DR_BOX
FOR I=21 TO 81 STEP 10
    BOX 80+I,80+I,80+300-I,80+300-I,5
NEXT
RETURN
EOP
GOSUB1
```

## GOTO

该指令用于跳到指定的标签。 标签长度不能超过 8 个字符。

### 指令语法

```
GOTO LABEL
```

```
:LABEL
```

参数	描述
LABEL	子程序的起点。标签的最大长度为 8 字符。

### 示例

```

DOWNLOAD "GOTO1.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 1
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
A=0
TOTAL=0
:START
  IF A<100 THEN
    GOTO SUM
  ELSE
    GOTO PRTOUT
ENDIF
:SUM
  A=A+1
  TOTAL=TOTAL+A
GOTO START
:PRTOUT
  B$="THE SUMMATION OF 1..100 IS "+STR$(TOTAL)
  TEXT 10,100,"3",0,1,1,B$
  PRINT 1
END
EOP

```

## INP\$( )

通过此函数由串口接收 1 比特的数据。

### 指令语法

INP\$(n)

参数	描述
n	1: 打印机的 com1 端口

## 示例

```

DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
T$=""
FOR I=1 TO 5
  T$=T$+INP$(1)
NEXT
TEXT 100,100,"4",0,1,1,"INP$(1)="+T$
PRINT 1
EOP
DEMO
12345

```

## REM

该指令用于程序注解。前缀为“REM”，打印机将忽略它。

## 指令语法

REM

## 示例

```

REM *****
REM This is a demonstration program*
REM *****
DOWNLOAD "REMARK.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 1
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 50,50,"3",0,1,1,"REMARK DEMO PROGRAM"
REM TEXT 50,100,"3",0,1,1,"REMARK DEMO PROGRAM"

```



```
PRINT 1,1
EOP
```

## OUT

该指令用于通过打印机串口发送数据。

### 指令语法

```
OUT “prompt” , variable
```

参数	描述
prompt	LCD 屏上显示的提示
variable	输出的信息

### 示例

```
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
PRICE$="123456"
OUT "PRICE:",PRICE$
EOP
```

## GETKEY( )

该指令用于获取 PAUSE 和 FEED 键的状态。指令会等到按下任意键之时，如果按下 PAUSE 键则返回 0，如果按下 FEED 键则返回 1。

### 指令语法

```
GETKEY( )
```

### 示例

```
DOWNLOAD "DEMO4.BAS"
SIZE 4,4
GAP 0,0
CLS
:START
A=GETKEY()
```

```
IF A=0 THEN GOTO PAUSEB
IF A=1 THEN GOTO FEEDB
:PAUSEB
CLS
TEXT 50,10,"4",0,1,1,"PAUSE key is pressed !"
PRINT 1
GOTO START
:FEEDB
CLS
TEXT 50,10,"4",0,1,1,"FEED key is pressed !"
PRINT 1
EOP
```

## INT( )

此函数将缩短浮动点数。

### 指令语法

INT(n)

#### 参数

#### 描述

n 正数或负数整数，浮点数或数学运算式

### 示例

```
DOWNLOAD "DEMO.BAS"
SIZE 4,2
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
INPUT "Number:",Num
CLS
REM ***** To round up or down*****
N=INT(Num+0.5)
IF N>Num THEN
    TEXT 50,100,"3",0,1,1,"To round up= "+STR$(N)
ELSE
    TEXT 50,100,"3",0,1,1,"To round down= "+STR$(N)
ENDIF
PRINT 1
EOP
```

## LEFT\$ ( )

此函数用于回传指定的字符数（从字符串的初始字符开始算起）。

### 指令语法

LEFT\$ (X\$, n)

参数	描述
X\$	要处理的字符串
n	返回的字符数

### 示例

```

DOWNLOAD "STR1.BAS"
SIZE 3.00,3.00
GAP 0.08,0.00
SPEED 4.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
A$="BARCODE PRINTER DEMO PRINTING"
C$=LEFT$(A$,10)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,100,"3",0,1,1,"10 LEFT 10 CHARS: "+C$
PRINT 1
EOP

```

## LEN ( )

此函数返回字符串的长度。

### 指令语法

LEN (string)

参数	描述
字符串	要量测的字符串长度

### 示例

```

DOWNLOAD "DEMO.BAS"
SIZE 3.00,3.00
GAP 0.08,0.00
SPEED 4.0
DENSITY 8

```

```

SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
A$="TSC Auto ID Technology Co., Ltd."
B=LEN(A$)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,50,"3",0,1,1,"STRING LENGTH="+STR$(B)
PRINT 1
EOP

```

## MID\$( )

此函数用于回传指定的字符数（从字符串的第 m 个字符开始算起）。

### 指令语法

MID\$(string, m, n)

参数	描述
string	要处理的字符串
m	字符串中第 m 个字符的起点 1 <= m <= 字符串长度
n	要回传的字符数

### 示例

```

DOWNLOAD "DEMO.BAS"
SIZE 3.00,3.00
GAP 0.08,0.00
SPEED 4.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
A$="TSC Auto ID Technology Co., Ltd."
E$=MID$(A$,11,10)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,200,"3",0,1,1,"10 MIDDLE CHARS: "+E$
PRINT 1
EOP

```

## RIGHT\$( )

此函数用于回传指定的字符数（从字符串的结尾开始算起）。

### 指令语法

RIGHT\$(X\$, n)

参数	描述
X\$	要处理的字符串
n	从字符串的最右边（结尾）回传的字符数

### 示例

```

DOWNLOAD "DEMO.BAS"
SIZE 3.00,3.00
GAP 0.08,0.00
SPEED 4.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
A$="TSC Auto ID Technology Co., Ltd."
D$=RIGHT$(A$,10)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,150,"3",0,1,1,"10 RIGHT CHARS: "+D$
PRINT 1
EOP

```

## STR\$( )

此函数将指定的值或表达式转换为相应的字符串。

### 指令语法

STR\$(n)

参数	描述
n	整数，浮点数或数学运算式

### 示例

```

DOWNLOAD "DEMO.BAS"
SIZE 3.00,3.00
GAP 0,0.00
SPEED 4.0
DENSITY 8

```

```
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
A$="TSC Auto ID Technology Co., Ltd."
F=100
G=500
H$=STR$(F+G)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,60,"3",0,1,1,"F="+STR$(F)
TEXT 10,110,"3",0,1,1,"G="+STR$(G)
TEXT 10,160,"3",0,1,1,"F+G="+H$
PRINT 1
EOP
DEMO
```

## STRCOMP ( )

根据字符串比较的结果，返回-1，0 或 1。

### 指令语法

STRCOMP (str1\$, str2\$ [, comp] )

参数	描述
str1\$	必要的。任何有效的字符串表达。
str2\$	必要的。任何有效的字符串表达。
comp	可选。指定字符串对比的类型。 0: 二进制对比。默认 1: 文本对比。对比不分大小写。

条件	返回值
str1\$在 str2\$之前	-1
str1\$等于 str2\$	0
str1\$在 str2\$之后	1

### 示例

```
DOWNLOAD "TEST.BAS"
STR1$ = "ABCD"
STR2$ = "abcd"
```

```
SIZE 4,1
GAP 0,0
CLS
```

```
TEXT 100,10,"3",0,1,1,STR$(result2) + ":" + STR1$ + " sorts before " + STR2$
TEXT 100,60,"3",0,1,1,STR$(result1) + ":" + STR1$ + " and " + STR2$ + " sort equally"
TEXT 100,110,"3",0,1,1,STR$(result3) + ":" + STR2$ + " sorts after " + STR1$
PRINT 1
EOP
TEST
```

## INSTR ( )

返回一个整数值用于指定首次发出的字串的起点位置。

### 指令语法

INSTR ([start,] str1\$, str2\$)

参数	描述
start	可选参数。数值表示法，用于设置每个搜索的起点。若省略，则在第一个字符的位置开始搜索。开始索引为 1。
str1\$	必要值。正在搜索字串表达式。
str2\$	必要值。已搜索到字串表达式。

### 示例

```
DOWNLOAD "DEMO.BAS"
string$ = "ABC123ABC123"
searchfor$ = "123"
starpos = 8
```

```
temp1 = INSTR(string$,searchfor$)
temp2 = INSTR(starpos,string$,searchfor$)
```

```
str1$ = searchfor$+" in "+string$+" is " + STR$(temp1)
str2$ = searchfor$+" in "+string$+" after " + STR$(starpos) + " is " + STR$(temp2)
```

```
SIZE 4,1
GAP 0,0
CLS
TEXT 10,10,"3",0,1,1,str1$
TEXT 10,60,"3",0,1,1,str2$
PRINT 1
EOP
DEMO
```

## TRIM\$ ( )

删除字符串首尾的空白格或特定字符。

### 指令语法

TRIM\$ (str\$ [, list\$])

参数	描述
str\$	要删减的字符串。
list\$	可选参数。要删除的字符。

### 示例

```
DOWNLOAD "DEMO.BAS"
```

```
data1$=" 1234567 "
```

```
data2$="a1234567a"
```

```
data3$="[<12345>]"
```

```
SIZE 4,1.5
```

```
GAP 0,0
```

```
CLS
```

```
TEXT 50,025,"2",0,1,1,LTRIM$(data1$)
```

```
TEXT 50,050,"2",0,1,1,TRIM$(data1$)
```

```
TEXT 50,075,"2",0,1,1,RTRIM$(data1$)
```

```
TEXT 50,100,"2",0,1,1,LTRIM$(data2$, "a")
```

```
TEXT 50,125,"2",0,1,1,TRIM$(data2$, "a")
```

```
TEXT 50,150,"2",0,1,1,RTRIM$(data2$, "a")
```

```
TEXT 50,175,"2",0,1,1,LTRIM$(data3$, "<>")
```

```
TEXT 50,200,"2",0,1,1,TRIM$(data3$, "<>")
```

```
TEXT 50,225,"2",0,1,1,RTRIM$(data3$, "<>")
```

```
PRINT 1
```

```
EOP
```

```
DEMO
```

## LTRIM\$ ( )

删除字符串前端的空白格。

### 指令语法

LTRIM\$ (str\$ [, list\$])

参数	描述
str\$	要删减的字符串。
list\$	可选参数。要删除的字符。



### 示例

```
DOWNLOAD "DEMO.BAS"
data1$=" 1234567 "
data2$="a1234567a"
data3$="[<12345>]"
```

```
SIZE 4,1.5
GAP 0,0
CLS
TEXT 50,025,"2",0,1,1,LTRIM$(data1$)
TEXT 50,050,"2",0,1,1,TRIM$(data1$)
TEXT 50,075,"2",0,1,1,RTRIM$(data1$)
TEXT 50,100,"2",0,1,1,LTRIM$(data2$,"a")
TEXT 50,125,"2",0,1,1,TRIM$(data2$,"a")
TEXT 50,150,"2",0,1,1,RTRIM$(data2$,"a")
TEXT 50,175,"2",0,1,1,LTRIM$(data3$,"[<>]")
TEXT 50,200,"2",0,1,1,TRIM$(data3$,"[<>]")
TEXT 50,225,"2",0,1,1,RTRIM$(data3$,"[<>]")
PRINT 1
EOP
DEMO
```

## RTRIM\$ ( )

删除字符串末端的空白格。

### 指令语法

RTRIM\$ (str\$ [, list\$])

参数	描述
str\$	要删减的字符串。
list\$	可选参数。要删除的字符。

### 示例

```
DOWNLOAD "DEMO.BAS"
data1$=" 1234567 "
data2$="a1234567a"
data3$="[<12345>]"
```

```
SIZE 4,1.5
GAP 0,0
CLS
TEXT 50,025,"2",0,1,1,LTRIM$(data1$)
```

```
TEXT 50,050,"2",0,1,1,TRIM$(data1$)
TEXT 50,075,"2",0,1,1,RTRIM$(data1$)
TEXT 50,100,"2",0,1,1,LTRIM$(data2$, "a")
TEXT 50,125,"2",0,1,1,TRIM$(data2$, "a")
TEXT 50,150,"2",0,1,1,RTRIM$(data2$, "a")
TEXT 50,175,"2",0,1,1,LTRIM$(data3$, "[<>]")
TEXT 50,200,"2",0,1,1,TRIM$(data3$, "[<>]")
TEXT 50,225,"2",0,1,1,RTRIM$(data3$, "[<>]")
PRINT 1
EOP
DEMO
```

## VAL ( )

此函数将数字字符转换为相应的整数或浮点数。

### 指令语法

VAL ("numeric character")

参数	描述
numeric character	“0~9” , ” . ”

### 示例

```
DOWNLOAD "DEMO.BAS"
SIZE 3.00,3.00
GAP 0.00,0.00
SPEED 4.0
DENSITY 8
SET CUTTER OFF
DIRECTION 0
REFERENCE 0,0
A$="TSC Auto ID Technology Co., Ltd."
F$="100"
G$="500"
CLS
H=VAL(F$)+VAL(G$)
I$=STR$(H)
TEXT 10,10,"3",0,1,1,A$
TEXT 10,60,"3",0,1,1,"F="+F$
TEXT 10,110,"3",0,1,1,"G="+G$
TEXT 10,160,"3",0,1,1,"F+G="+I$
PRINT 1
EOP
DEMO
```

## 6. 设备重置指令

### SET COUNTER

计数器可以是实时的或者一个变量。该命令在程序里设置计数器的序号和其增量。有三种类型的计数器：数字（0~9~0），小写字母（a~z~a），或大写字母（A~Z~A）。

#### 指令语法

SET COUNTER @n step

@n = "Expression"

#### 参数

#### 描述

@n	n: 计数器序号。打印机里共有 51 个可变计数器（@0~@50）
step	计数间隔，可为正数或负数。 -999999999 <= step <= 999999999 如果计数器作为一个固定变量，请将间隔数设为 0。
Expression	初始字串。字串长度为 101 比特。

#### 示例

```
SIZE 3,3
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET COUNTER @1 1
@1="00001"
SET COUNTER @2 5
@2="AB000001"
CLS
TEXT 50,50,"3",0,1,1,@1
BARCODE 50,100,"39",48,1,0,2,4,@2
PRINT 2,1
```

### SET CUTTER

该指令用于设置激活或禁用切刀功能，并定义一次要裁切多少个标签。关闭电源后此设置将被保存在打印机内存中。

#### 指令语法

SET CUTTER OFF/BATCH/pieces

参数	描述
OFF	禁用切刀功能
BATCH	设置打印机在打印后切纸
Pieces	设置每次裁切标签的数量 $0 \leq \text{pieces} \leq 65535$

### 示例

```

REM ***SET CUTTER FUNCTION OFF EXAMPLE PROGRAM***
SIZE 3,3
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 50,50,"3",0,1,1,"SET CUTTER OFF"
PRINT 3
REM ***This program cuts once at the batch***
SET CUTTER BATCH
CLS
TEXT 50,50,"3",0,1,1,"SET CUTTER BATCH"
PRINT 3,2
REM ***This program cuts every label***
SET CUTTER 1
CLS
TEXT 50,50,"3",0,1,1,"SET CUTTER 1"
PRINT 3,2
REM ***This program cuts 2 label***
SET CUTTER 2
CLS
TEXT 50,50,"3",0,1,1,"SET CUTTER 2"
PRINT 3,2

```

## SET PARTIAL\_CUTTER

该指令用于设置激活或禁用切刀功能，并定义一次要裁切多少个标签。关闭电源后此设置将被保存在打印机内存中。此功能防止标签在裁切后回退。

### 指令语法

```
SET PARTIAL_CUTTER OFF/BATCH/pieces
```

参数	描述
OFF	禁用切刀功能
BATCH	设置打印机在打印后切纸
Pieces	设置每次裁切标签的数量 $0 \leq \text{pieces} \leq 65535$

### 示例

```
REM **SET PARTIAL_CUTTER FUNCTION OFF EXAMPLE PROGRAM**
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET PARTIAL_CUTTER OFF
CLS
TEXT 50,50,"3",0,1,1,"SET PARTIAL_CUTTER OFF"
PRINT 3
REM ***This program cuts once at the batch***
SET PARTIAL_CUTTER BATCH
CLS
TEXT 50,50,"3",0,1,1,"SET PARTIAL_CUTTER BATCH"
PRINT 3,2
REM ***This program cuts every label***
SET PARTIAL_CUTTER 1
CLS
TEXT 50,50,"3",0,1,1,"SET PARTIAL_CUTTER 1"
PRINT 3,2
REM ***This program cuts 2 label***
SET PARTIAL_CUTTER 2
CLS
TEXT 50,50,"3",0,1,1,"SET PARTIAL_CUTTER 2"
PRINT 3,2
```

## SET BACK

此设置用于 SET CUTTER 功能之后。该功能防止标签在裁切后回退。

### 指令语法

SET BACK OFF/ON

参数	描述
OFF	禁用此功能
ON	启用此功能

### 示例

```
REM **SET BACK FUNCTION OFF EXAMPLE PROGRAM**
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 1
REFERENCE 0,0
SET CUTTER 1
SET BACK OFF
CLS
TEXT 50,50,"3",0,1,1,"SET BACK OFF"
PRINT 3
CLS
SET CUTTER 1
SET BACK ON
TEXT 50,50,"3",0,1,1,"SET BACK ON"
PRINT 3
```

## SET KEY1, SET KEY2, SET KEY3

此设置用于启用/禁用 KEY1/KEY2/KEY3 功能。KEY1 的默认为“MENU（菜单）”键，KEY2 默认为“PAUSE（暂停）”键，KEY3 默认为“FEED”键。在设置 KEY1/KEY2/KEY3 功能之前，先禁用 KEY1/KEY2/KEY3。即使打印机关机，此设置仍然会保留在打印机里。

### 指令语法

```
SET KEY1 ON/OFF
SET KEY2 ON/OFF
SET KEY3 ON/OFF
```

参数	描述
ON	启用 KEY $n$ 功能
OFF	禁用 KEY $n$ 功能

### 示例

```
DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET KEY1 OFF
```

```

SET KEY2 OFF
SET KEY3 OFF
KEY1=0
KEY2=0
KEY3=0

:START
IF KEY1=1 THEN
CLS
TEXT 100,10,"3",0,1,1,"KEY1 (MENU key) is pressed!! "
PRINT 1,1
ELSEIF KEY2=1 THEN
CLS
TEXT 100,10,"3",0,1,1,"KEY2 (PAUSE key) is pressed!!"
PRINT 1,1
ELSEIF KEY3=1 THEN
CLS
TEXT 100,10,"3",0,1,1,"KEY3 (FEED key) is pressed!!"
TEXT 100,60,"3",0,1,1,"End of test"
PRINT 1,1
SET KEY1 ON
SET KEY2 ON
SET KEY3 ON
END
ENDIF
GOTO START
EOP
DEMO

```

## SET LED1, SET LED2, SET LED3

此设置用于控制 LED 开/关功能。

### 指令语法

```

SET LED1 ON/OFF
SET LED2 ON/OFF
SET LED3 ON/OFF

```

<u>LED 序号</u>	<u>默认功能</u>
LED1	开机/关机
LED2	打印机在线/脱机
LED3	错误/正常

<u>参数</u>	<u>描述</u>
-----------	-----------

ON        启用 LED $n$  功能  
OFF       禁用 LED $n$  功能

### 示例

```
DOWNLOAD "DEMO4.BAS"
SET LED1 OFF
SET LED2 OFF
SET LED3 OFF
FOR I=1 TO 100
  LED1=0
  LED2=0
  LED3=0
  IF I-INT(I/2)*2=0 THEN
    LED1=1
  ELSEIF I-INT(I/3)*3=0 THEN
    LED2=1
  ELSE
    LED3=1
  ENDIF
NEXT
LED1=1
LED2=1
LED3=0
SET LED1 ON
SET LED2 ON
SET LED3 ON
EOP
DEMO4
```

## SET PEEL

此设置用于启用/禁用自动剥离功能。默认设置为关。当设置此功能为开，打印机将在打印完每张标签后停止，直到剥离的标签被取走后才继续打印下一张标签。关机后此设置仍然会保存在打印机存储器里。

### 指令语法

SET PEEL ON/OFF

参数	描述
ON	启用自动剥离功能
OFF	禁用自动剥离功能



### 示例

```
REM ***SELF-PEELING FUNCTION ON***
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL ON
CLS
TEXT 50,100,"3",0,1,1,"SELF-PEELING FUNCTION TEST"
PRINT 5
```

## SET TEAR & SET STRIPER

该指令用于启用/禁用撕纸后将标签定位到缝标/黑标位置的功能。关机后此设置仍然会保存在打印机存储器里。

### 参数

### 描述

ON	标签将在打印后定位到撕纸位置。
OFF	标签不会在打印后定位到撕纸位置。标签的前端与打印头平齐。

### 示例

```
REM ***TEAR FUNCTION ON***
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET TEAR ON
CLS
TEXT 50,100,"3",0,1,1,"TEAR FUNCTION TEST"
PRINT 1
```

## SET GAP

此设置用于设置缝标传感器的灵敏度。开机时按下 **PAUSE** 键，打印机将启动自动缝标传感器校准。如果底纸的厚度和带有底纸的标签的厚度与传感器没有明显差异，或者标签上有预印的标记或图案时，此功能会暂停工作。在这种情况下，用户必须通过此命令手动校准缝标传感器。关机后此设置仍然会保存在打印机存储器里。

## 指令语法

SET GAP n/AUTO/OFF/0,/REVERSE/OBVERSE

参数	描述
N	缝标传感器光发射强度。0 为最低值。
AUTO	打印机进纸 2 到 3 张标签来校验缝标。如果标签是连续性的，打印机将进纸 10~20 英寸以确认标签是否是连续的。
OFF	禁用 SET GAP AUTO 功能。
0,	自动校准缝标尺寸。
REVERSE	当黑标为标签正面的分隔标志且未被黑标(Black Mark)传感器侦测到时启用此功能。可通过缝标(GAP)传感器侦测到的部分将被定义为可打印区域，否则将被定义为介质的缝标(GAP)。
OBVERSE	禁用“SET GAP REVERSE”功能

## 示例

以下示例是在 DOS 环境下操作的，通过并口连接来设置标签尺寸、缝标距离和传感器灵敏度。

```
C:\>COPY CON LPT1<ENTER>
SIZE 4,2.5<ENTER>
GAP 0.12,0<ENTER>
SET GAP 1<ENTER>
<CTRL><Z><ENTER>
C:\>
```

备注：<ENTER>表示键盘上的“ENTER”键。<CTRL>表示按住键盘上的“CTRL”键。

## 解决方案：

按 FEED 键进行测试。打印机是否在错误灯没闪烁时停在每个标签的相同位置？如果不是，则将设置调整为更大的数字。当你调整此设置时，从 0 开始，然后递增到更高值。

## SET RIBBON

此设置用于启用/禁用碳带传感器侦测。（热转印/直接热敏）打印机开机后，它会侦测到碳带该用直接热敏打印还是热转印打印。此设置不会被保存在打印机存储器里。

## 指令语法

SET RIBBON ON /OFF

参数	描述
ON	热转印打印
OFF	直接热敏打印

## 示例

```
REM ***Direct printing****
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET RIBBON OFF
CLS
BARCODE 100,100,"39",48,1,0,2,5,"CODE 39"
PRINT 1
```

## SET COM1

此设置用于定义打印机串口的通讯参数。

### 指令语法

SET COM1 baud, parity, data, stop

参数	描述
baud	<p>可用的波特率如下：</p> <p>24: 2400 bps</p> <p>48: 4800 bps</p> <p>96: 9600 bps</p> <p>19: 19200 bps</p> <p>38: 38400 bps</p> <p>57: 57600 bps</p> <p>115: 115200 bps</p>
parity	<p>奇偶校验</p> <p>N: 无</p> <p>E: 奇数校验</p> <p>O: 偶数校验</p>
Data	<p>数据位</p> <p>8: 8 位数据</p> <p>7: 7 位数据</p>
stop	<p>停止位</p> <p>1: 1 停止位</p> <p>2: 2 停止位</p>

## 示例

以下示例中，并口通过 MS-DOS 模式来设置打印机串口。

```
C:\>COPY CON LPT1<ENTER>
SET COM1 19,N,8,1<ENTER>
<CTRL><Z><ENTER>
C:\>
```

**备注：**<ENTER>表示键盘上的“ENTER”键。<CTRL><Z>表示同时按住键盘上的“CTRL”键和<Z>键。

## SET PRINTKEY

该指令用于打印一张标签然后定位到撕纸的位置。按下 FEED 键打印下一张标签或批量标签。如果标签内容包括串行文本或条码，它将相应更改序列号。该设置会保存在打印机存储器里。

### 指令语法

SET PRINTEKY OFF/ON/AUTO/<num>

参数	描述
OFF	禁用此功能
ON	启用此功能
AUTO	启用此功能
<num>	按下 FEED 键打印的标签数目

## 示例

```
Execute:
SIZE 4,2.5
GAP 0.12,0
SET PRINTKEY ON
SET COUNTER @0 1
@0="0001"
CLS
TEXT 10,10,"5",0,1,1,@0
PRINT 1
```

## SET REPRINT

该指令用于设置发生“缺纸”、“缺碳带”或“开盖”错误之后的重打功能。

### 指令语法

SET REPRINT OFF/ON

参数	描述
OFF	禁用此功能
ON	启用此功能

#### 示例

```
SET REPRINT ON
```

## PEEL

该指令用于获取剥离传感器的状态。属性为只读。

#### 指令语法

PEEL

返回值	描述
0	纸张不在剥离传感器上方
1	纸张在剥离传感器上方

#### 示例

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
SPEED 4
DENSITY 8
SET PEEL OFF
SET KEY1 OFF
SET LED1 OFF
SET LED3 OFF
:START
LED1=0
LED3=0
  IF KEY1=1 THEN GOTO A
GOTO START
:A
LED1=1
CLS
TEXT 10,10,"3",0,1,1,"PEEL Function Test!! "
PRINT 1,1
:B
LED1=0
IF PEEL=1 THEN
LED3=1
GOTO B
ELSE
```

```
CLS
TEXT 10,10,"3",0,1,1,"The label is removed from the PEEL sensor!!"
PRINT 1,1
GOTO START
ENDIF
EOP
DEMO
```

## LED1, LED2, LED3

该指令用于设置 LED 开/关。属性为只写入。指定 1 为 LED 开，0 为 LED 关。使用该指令之前，确保已经取消 LED 默认设置。请参阅 SET LED 指令。

### 指令语法

LEDm = n

参数	描述
m	m=1, LED1 m=2, LED2 m=3, LED3
n	0: 关掉 LED 1: 打开 LED

### 示例

```
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.12,0
SPEED 4
DENSITY 8
DIRECTION 1
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET LED1 OFF
SET LED2 OFF
SET LED3 OFF
LED1=0
LED2=1
LED3=0
EOP
```

## KEY1, KEY2, KEY3

该指令用于读取 KEY1，KEY2 和 KEY3 的状态。

### 指令语法

KEYm = n

<u>Key</u>	<u>Return Value</u>
KEY1 (MENU)	0: 释放按键 1: 按下按键
KEY1 (PAUSE)	0: 释放按键 1: 按下按键
KEY1 (FEED)	0: 释放按键 1: 按下按键

### 示例

```

DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
SPEED 4
DENSITY 8
DIRECTION 1
REFERENCE 0,0
SET LED1 OFF
SET KEY1 OFF
LED1=0
:START
IF KEY1=1 THEN
LED1=1
CLS
TEXT 100,10,"3",0,1,1,"KEY FUNCTION TEST"
PRINT 1,1
ELSE
LED1=0
ENDIF
GOTO START
EOP
DEMO

```

## 7. 打印机全域变数

### @LABEL

这个变数用于计算已打印了多少张标签。如果打印机重置，这个属性无法初始化，但如果打印机关机，它将会被保留。

#### 指令语法

Write attribute: @LABEL=n or @LABEL= "n"

Read attribute: A=LABEL or A\$=STR\$(LABEL)

参数	描述
n	打印的标签数。 0<=n<=999999999

#### 示例

```

DOWNLOAD "DEMO.BAS"
SIZE 4,2.5
GAP 2 mm,0
SPEED 6
DENSITY 12
CLS
TEXT 10,50,"3",0,1,1,@LABEL
TEXT 10,100,"3",0,1,1,"@LABEL="+STR$(LABEL)
TEXT 10,150,"3",0,1,1,"*****Statement 1*****"
IF LABEL>1000 THEN
    TEXT 10,200,"3",0,1,1,"LABEL>1000"
ELSE
    TEXT 10,200,"3",0,1,1,"LABEL<1000"
ENDIF
TEXT 10,250,"3",0,1,1,"*****Statement 1*****"
A=LABEL
IF A>1000 THEN
    TEXT 10,300,"3",0,1,1,"A>1000"
ELSE
    TEXT 10,300,"3",0,1,1,"A<1000"
ENDIF
TEXT 10,350,"3",0,1,1,"*****Statement 3*****"
A$=STR$(LABEL)
IF VAL(A$)>1000 THEN
    TEXT 10,400,"3",0,1,1,"VAL(A$)>1000"
ELSE
    TEXT 10,400,"3",0,1,1,"VAL(A$)<1000"
ENDIF

```



```
PRINT 1,1
EOP
```

## YEAR

此变数通过实时时钟（RTC）来读取/写入年份数据。RTC 支持四位数的年份格式。

### 指令语法

```
Write attribute: YEAR = 02
Read attribute: A = YEAR
Range: 00~50 = 2000~2050; 51~99 = 1951~1999
```

### 示例

```
DOWNLOAD "SetYear.BAS"
REM *****Set Year Parameter to RTC*****
YEAR=05
EOP
SetYear

DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS

REM *****Read YEAR parameter form RTC*****
YEAR$=STR$(YEAR)
Y=YEAR
REM *****Print*****
TEXT 10,10,"5",0,1,1,"YEAR1="+YEAR$
TEXT 10,110,"5",0,1,1,"YEAR2="+STR$(Y)
TEXT 10,210,"5",0,1,1,"YEAR3="+STR$(YEAR)
PRINT 1
EOP
DEMO
```

## MONTH

此变数通过实时时钟（RTC）来读取/写入月份数据。RTC 支持两位数（01~12）的月份格式。

### 指令语法

Write attribute: MONTH = 01

Read attribute: A = MONTH

Range: 01~12

### 示例

```
DOWNLOAD "SetMonth.BAS"
```

```
REM *****Set Month Parameter to RTC*****
```

```
MONTH=05
```

```
EOP
```

```
SetMonth
```

```
DOWNLOAD "DEMO.BAS"
```

```
SIZE 3,3
```

```
GAP 0.08,0
```

```
DENSITY 8
```

```
SPEED 4
```

```
DIRECTION 0
```

```
REFERENCE 0,0
```

```
SET CUTTER OFF
```

```
SET PEEL OFF
```

```
CLS
```

```
REM *****Read Month parameter form RTC*****
```

```
MONTH$=STR$(MONTH)
```

```
M=MONTH
```

```
REM *****Print*****
```

```
TEXT 10,10,"5",0,1,1,"MONTH1="+MONTH$
```

```
TEXT 10,110,"5",0,1,1,"MONTH2="+STR$(M)
```

```
TEXT 10,210,"5",0,1,1,"MONTH3="+STR$(MONTH)
```

```
PRINT 1
```

```
EOP
```

```
DEMO
```

## DATE

此变数通过实时时钟（RTC）来读取/写入日期数据。RTC 支持两位数（01~31）的日期格式。

### 指令语法

Write attribute: DATE = 12

Read attribute: A = DATE

Range: 01~31

### 示例

DOWNLOAD "SetDate.BAS"

REM \*\*\*\*\*Set Date Parameter to RTC\*\*\*\*\*

DATE=30

EOP

SetDate

DOWNLOAD "DEMO.BAS"

SIZE 3,3

GAP 0.08,0

DENSITY 8

SPEED 4

DIRECTION 0

REFERENCE 0,0

SET CUTTER OFF

SET PEEL OFF

CLS

REM \*\*\*\*\*Read Date parameter form RTC\*\*\*\*\*

DATE\$=STR\$(DATE)

D=DATE

REM \*\*\*\*\*Print\*\*\*\*\*

TEXT 10,10,"5",0,1,1,"DATE1="+DATE\$

TEXT 10,110,"5",0,1,1,"DATE2="+STR\$(D)

TEXT 10,210,"5",0,1,1,"DATE3="+STR\$(DATE)

PRINT 1

EOP

DEMO

## WEEK

此变数通过实时时钟（RTC）来读取/写入星期数据，以个位数来表示（1~7）。

### 指令语法

Write attribute: WEEK = 3

Read attribute: A = WEEK

Range: 1(Sunday)~7(Saturday)

### 示例

DOWNLOAD "SetWeek.BAS"

REM \*\*\*\*\*Set Week Parameter to RTC\*\*\*\*\*

WEEK=6

EOP

SetWeek

DOWNLOAD "DEMO.BAS"

SIZE 3,3

GAP 0.08,0

DENSITY 8

SPEED 4

DIRECTION 0

REFERENCE 0,0

SET CUTTER OFF

SET PEEL OFF

CLS

REM \*\*\*\*\*Read Week parameter form RTC\*\*\*\*\*

WEEK\$=STR\$(WEEK)

W=WEEK

REM \*\*\*\*\*Print\*\*\*\*\*

TEXT 10,10,"5",0,1,1,"WEEK1="+WEEK\$

TEXT 10,110,"5",0,1,1,"WEEK2="+STR\$(W)

TEXT 10,210,"5",0,1,1,"WEEK3="+STR\$(WEEK)

PRINT 1

EOP

DEMO

## **HOUR**

此变数通过实时时钟（RTC）来读取/写入小时数据。RTC 支持两位数（00~23）的小时格式。

### **指令语法**

Write attribute: HOUR = 12

Read attribute: A = HOUR

Range: 00~23

### **示例**

DOWNLOAD "SetHour.BAS"

REM \*\*\*\*\*Set Hour Parameter to RTC\*\*\*\*\*

HOUR=11

EOP

SetHour

DOWNLOAD "DEMO.BAS"

SIZE 3,3

GAP 0.08,0

DENSITY 8

SPEED 4

DIRECTION 0

REFERENCE 0,0

SET CUTTER OFF

SET PEEL OFF

CLS

REM \*\*\*\*\*Read Hour parameter form RTC\*\*\*\*\*

HOUR\$=STR\$(HOUR)

H=HOUR

REM \*\*\*\*\*Print\*\*\*\*\*

TEXT 10,10,"5",0,1,1,"HOUR1="+HOUR\$

TEXT 10,110,"5",0,1,1,"HOUR2="+STR\$(H)

TEXT 10,210,"5",0,1,1,"HOUR3="+STR\$(HOUR)

PRINT 1

EOP

DEMO

## MINUTE

此变数通过实时时钟（RTC）来读取/写入分钟数据。RTC 支持两位数的分钟格式（00~59）。

### 指令语法

Write attribute: MINUTE = 12

Read attribute: A = MINUTE

Range: 00~59

### 示例

DOWNLOAD "SetMinute.BAS"

REM \*\*\*\*\*Set Minute Parameter to RTC\*\*\*\*\*

MINUTE=59

EOP

SetMinute

DOWNLOAD "DEMO.BAS"

SIZE 3,3

GAP 0.08,0

DENSITY 8

SPEED 4

DIRECTION 0

REFERENCE 0,0

SET CUTTER OFF

SET PEEL OFF

CLS

REM \*\*\*\*\*Read Minute parameter form RTC\*\*\*\*\*

MINUTE\$=STR\$(MINUTE)

MIN=MINUTE

REM \*\*\*\*\*Print\*\*\*\*\*

TEXT 10,10,"5",0,1,1,"MINUTE1="+MINUTE\$

TEXT 10,110,"5",0,1,1,"MINUTE2="+STR\$(MIN)

TEXT 10,210,"5",0,1,1,"MINUTE3="+STR\$(MINUTE)

PRINT 1

EOP

DEMO

## SECOND

此变数通过实时时钟（RTC）来读取/写入秒钟数据。RTC 支持两位数的秒钟格式（00~59）。

### 指令语法

Write attribute: SECOND = 12

Read attribute: A = SECOND

Range: 00~59

### 示例

DOWNLOAD "SetSecond.BAS"

REM \*\*\*\*\*Set Second Parameter to RTC\*\*\*\*\*

SECOND=59

EOP

SetSecond

DOWNLOAD "DEMO.BAS"

SIZE 3,3

GAP 0.08,0

DENSITY 8

SPEED 4

DIRECTION 0

REFERENCE 0,0

SET CUTTER OFF

SET PEEL OFF

CLS

REM \*\*\*\*\*Read Second parameter form RTC\*\*\*\*\*

SECOND\$=STR\$(SECOND)

SEC=SECOND

REM \*\*\*\*\*Print\*\*\*\*\*

TEXT 10,10,"5",0,1,1,"SECOND1="+SECOND\$

TEXT 10,110,"5",0,1,1,"SECOND2="+STR\$(SEC)

TEXT 10,210,"5",0,1,1,"SECOND3="+STR\$(SECOND)

PRINT 1

EOP

DEMO

## @YEAR

此变数通过实时时钟（RTC）来读取/写入年份数据。RTC 支持两位数的年份格式。不使用 BASIC 语言功能就可直接获取@YEAR 全域变数。

### 指令语法

Write attribute: @YEAR = "01"

Read attribute: @YEAR

Range: 00~99

### 示例

```
REM *****Set @YEAR*****
```

```
@YEAR="05"
```

```
REM *****Print*****
```

```
SIZE 3,3
```

```
GAP 0.08,0
```

```
DENSITY 8
```

```
SPEED 6
```

```
DIRECTION 0
```

```
REFERENCE 0,0
```

```
SET CUTTER OFF
```

```
SET PEEL OFF
```

```
CLS
```

```
TEXT 10,10,"5",0,1,1,"@YEAR"
```

```
TEXT 310,10,"5",0,1,1,@YEAR
```

```
PRINT 1
```

## @MONTH

此变数通过实时时钟（RTC）来读取/写入月份数据。RTC 支持两位数的月份格式。不使用 BASIC 语言功能就可直接获取@MONTH 全域变数。

### 指令语法

Write attribute: @MONTH = "01"

Read attribute: @MONTH

Range: 01~12

### 示例

```
REM *****Set @MONTH*****
```

```
@MONTH="12"
```

```
REM *****Print*****
```

```
SIZE 3,3
```



```
GAP 0.08,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"5",0,1,1,"@MONTH"
TEXT 310,10,"5",0,1,1,@MONTH
PRINT 1
```

## @DATE

此变数通过实时时钟（RTC）来读取/写入日期数据。RTC 支持两位数的日期格式。不使用 BASIC 语言功能就可直接获取@DATE 全域变数。

### 指令语法

Write attribute: @DATE = "12"

Read attribute: @DATE

Range: 01~31

### 示例

```
REM *****Set @DATE*****
@DATE="31"

REM *****Print*****
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"5",0,1,1,"@DATE"
TEXT 310,10,"5",0,1,1,@DATE
PRINT 1
```

## @DAY

此变数通过实时时钟（RTC）来读取/写入星期数据，以个位数（1~7）表示。不使用 BASIC 语言功能就可直接获取@DAY 全域变数。

### 指令语法

Write attribute: @DAY = "3"

Read attribute: @DAY

Range: 1(Sunday)~7(Saturday)

### 示例

```
REM *****Set @DAY*****
```

```
@DAY="5"
```

```
REM *****Print*****
```

```
SIZE 3,3
```

```
GAP 0.08,0
```

```
DENSITY 8
```

```
SPEED 6
```

```
DIRECTION 0
```

```
REFERENCE 0,0
```

```
SET CUTTER OFF
```

```
SET PEEL OFF
```

```
CLS
```

```
TEXT 10,10,"5",0,1,1,"@DAY"
```

```
TEXT 310,10,"5",0,1,1,@DAY
```

```
PRINT 1
```

## @HOUR

此变数通过实时时钟（RTC）来读取/写入小时数据。RTC 支持两位数（00~23）的小时格式。不使用 BASIC 语言功能就可直接获取@HOUR 全域变数。

### 指令语法

Write attribute: @HOUR = "12"

Read attribute: @HOUR

Range: 00~23

### 示例

```
REM *****Set @HOUR*****
```

```
@HOUR="23"
```

```
REM *****Print*****
```

```
SIZE 3,3
```

```
GAP 0.08,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"5",0,1,1,"@HOUR"
TEXT 310,10,"5",0,1,1,@HOUR
PRINT 1
```

## @MINUTE

此变数通过实时时钟（RTC）来读取/写入分钟数据。RTC 支持两位数（00~59）的分钟格式。不使用 BASIC 语言功能就可直接获取@MINUTE 全域变数。

### 指令语法

Write attribute: @MINUTE = "12"

Read attribute: @MINUTE

Range: 00~59

### 示例

```
REM *****Set @MINUTE*****
@MINUTE="59"

REM *****Print*****
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
CLS
TEXT 10,10,"5",0,1,1,"@MINUTE"
TEXT 310,10,"5",0,1,1,@MINUTE
PRINT 1
```

## @SECOND

此变数通过实时时钟（RTC）来读取/写入秒钟数据。RTC 支持两位数（00~59）的秒钟格式。不使用 BASIC 语言功能就可直接获取@SECOND 全域变数。

### 指令语法

Write attribute: @SECOND = "12"

Read attribute: @SECOND

Range: 00~59

### 示例

```
REM *****Set @SECOND*****
```

```
@SECOND = "59"
```

```
REM *****Print*****
```

```
SIZE 3,3
```

```
GAP 0,0
```

```
DENSITY 8
```

```
SPEED 6
```

```
DIRECTION 0
```

```
REFERENCE 0,0
```

```
SET CUTTER OFF
```

```
SET PEEL OFF
```

```
CLS
```

```
TEXT 10,10,"5",0,1,1,"@SECOND"
```

```
TEXT 310,10,"5",0,1,1,@SECOND
```

```
PRINT 1
```